ITOUG
ITALIAN ORACLE USER GROUP

Write Less (Code)
With More (Features)

Can you read this query easily?

If not, please sit closer

```
select department_id,
       count(*) number_of_employees,
       avg(salary) avg_salary
from employees
group by department_id
having max(salary) > 2 * min(salary)
```

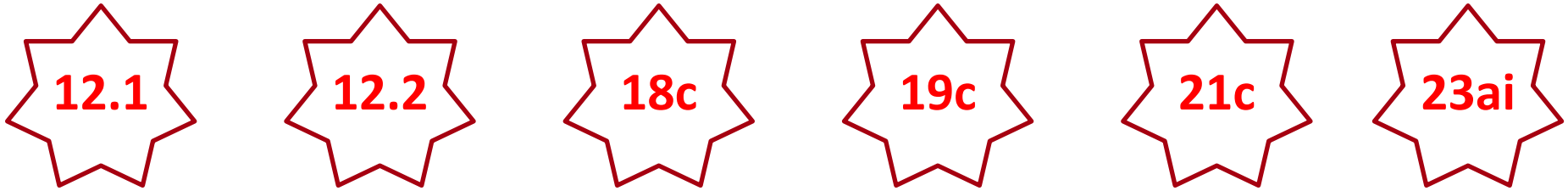

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

𝕏 @DBoriented

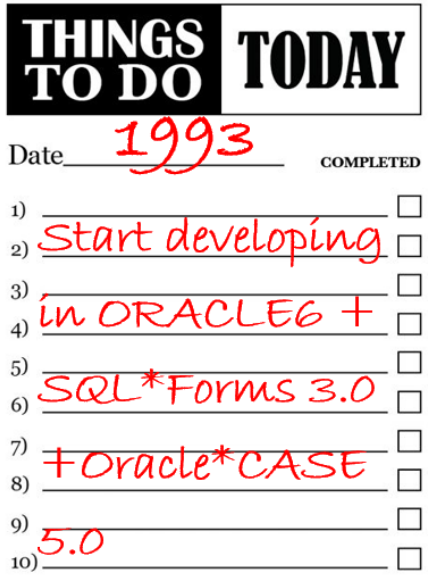
Write Less (Code)
With More (Features)

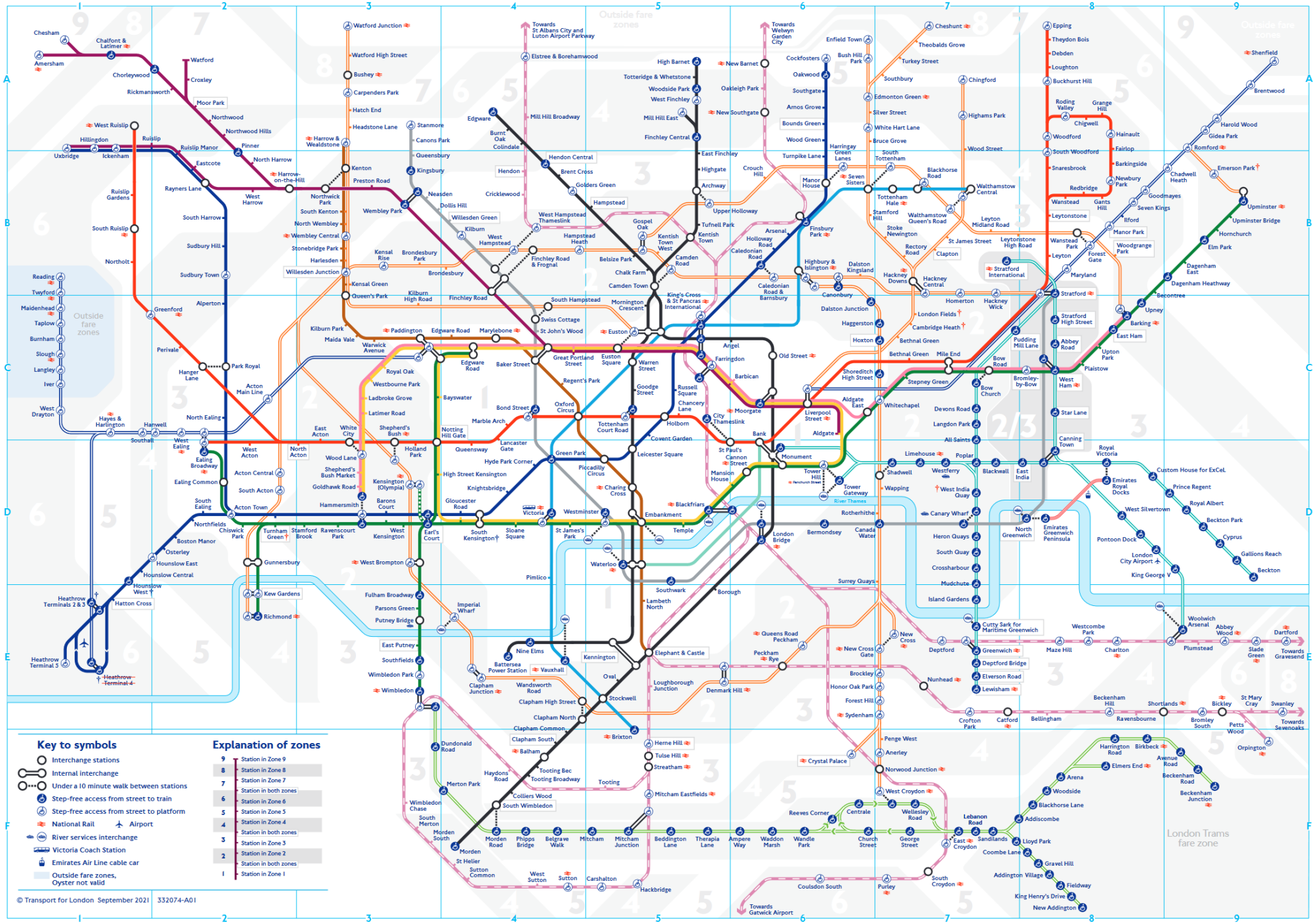


https://db-oriented.com

Follow @DBoriented

Who Am I?





Check before you travel

† **Heathrow**
TfL Rail customers should change at Terminals 2&3 for free rail transfer to Terminal 5.

† **Heathrow Terminal 4**
Closed until further notice.

† **Hounslow West**
Step-free access for manual wheelchairs only.

† **South Kensington**
Piccadilly line trains not stopping until spring 2022.

† Services or access at these stations are subject to variation.
To check before you travel, visit **tfl.gov.uk/plan-a-journey**

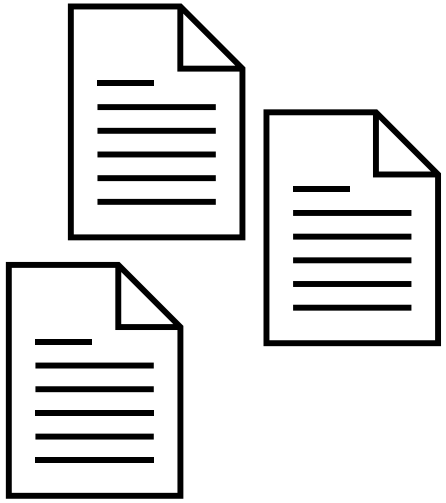
Key to lines

- Bakerloo
- Central
- Circle
- District
- Hammersmith & City
- Jubilee
- Metropolitan
- Northern
- Piccadilly
- Victoria
- Waterloo & City
- DLR
- Emirates Air Line cable car (special fares apply)
- London Overground
- TfL Rail
- Thameslink - Not operated by TfL. If you need assistance, please speak to staff at your origin station so this can be arranged at the destination station or check assisted travel at www.thameslinkrailway.co.uk
- London Trams
- District open weekends and on some public holidays

Demo – GTFS

- General Transit Feed Specification
- File Set
  - agency.txt
  - stops.txt
  - routes.txt
  - trips.txt
  - stop_times.txt
  - calendar.txt
  - calendar_dates.txt
  - ...
- London Underground
  - https://data.bus-data.dft.gov.uk/timetable/download/gtfs-file/london/

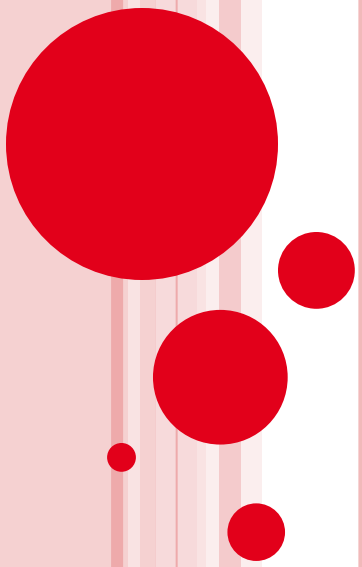
SELECT … FROM



External Tables

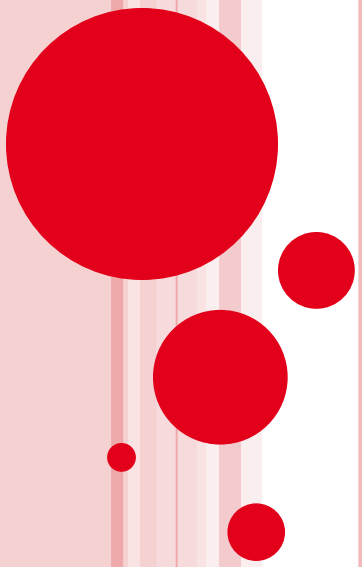
External Tables

- Ad-Hoc Queries
- Ongoing Loading
- Runtime Adjustments

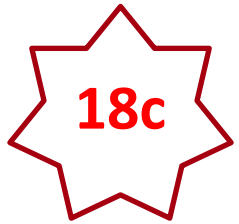


External Tables

- Ad-Hoc Queries
- Ongoing Loading
- Runtime Adjustments



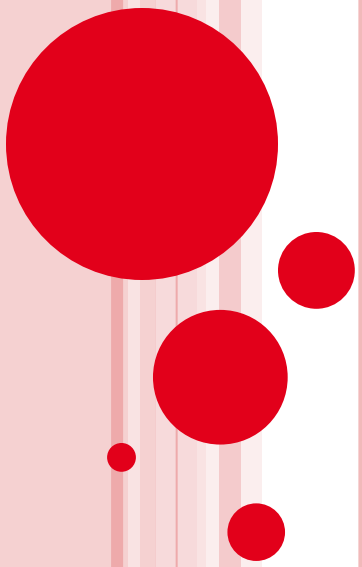
Inline External Tables



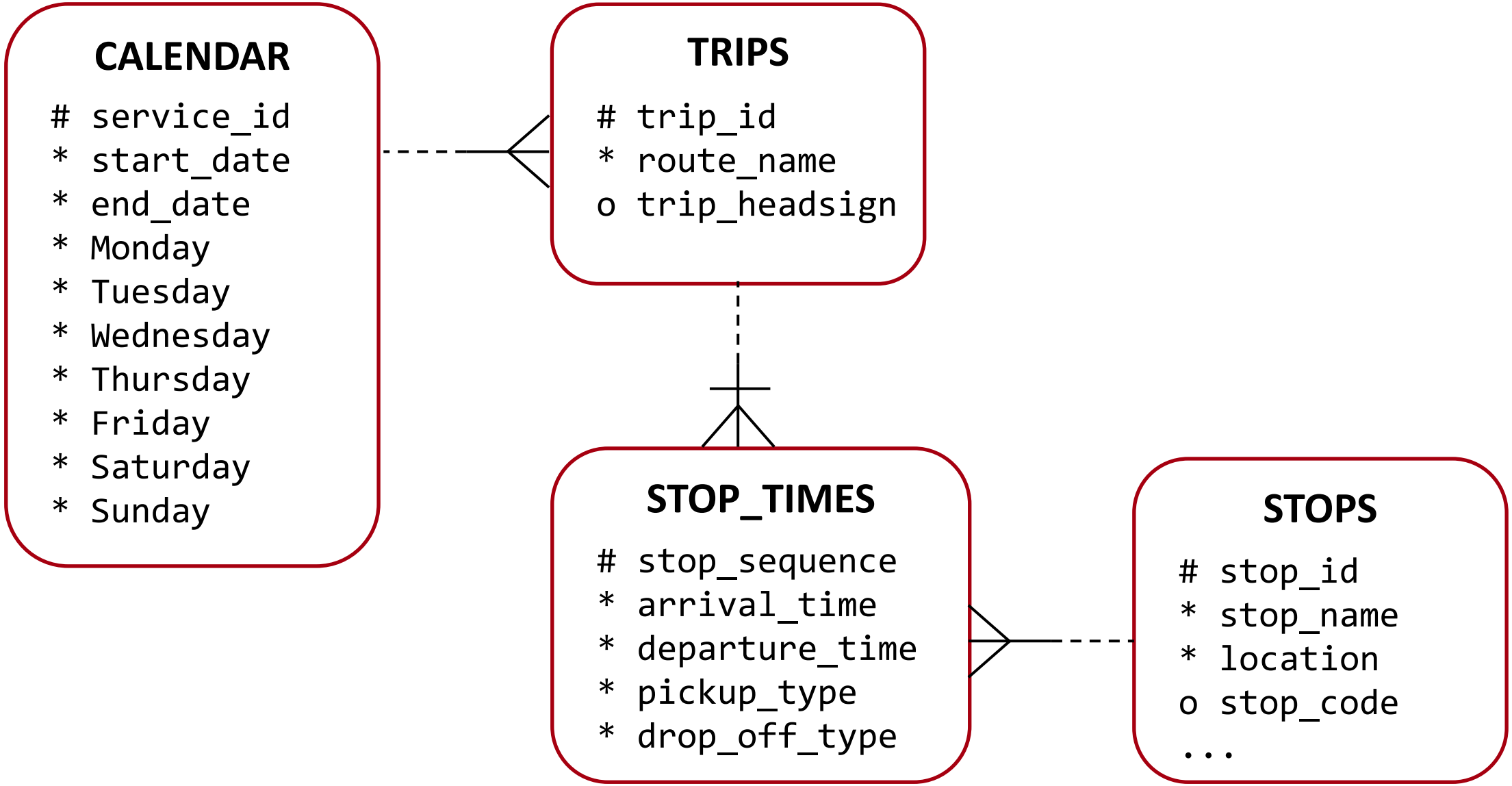
- Runtime definition
- No persistent objects in the data dictionary

External Tables

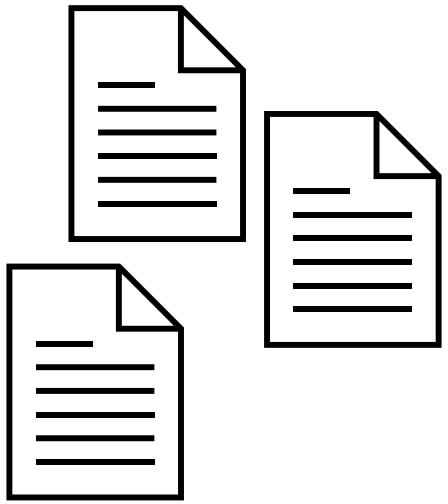
- ✓ Ad-Hoc Queries
- ➢ **Ongoing Loading**
- ➢ Runtime Adjustments

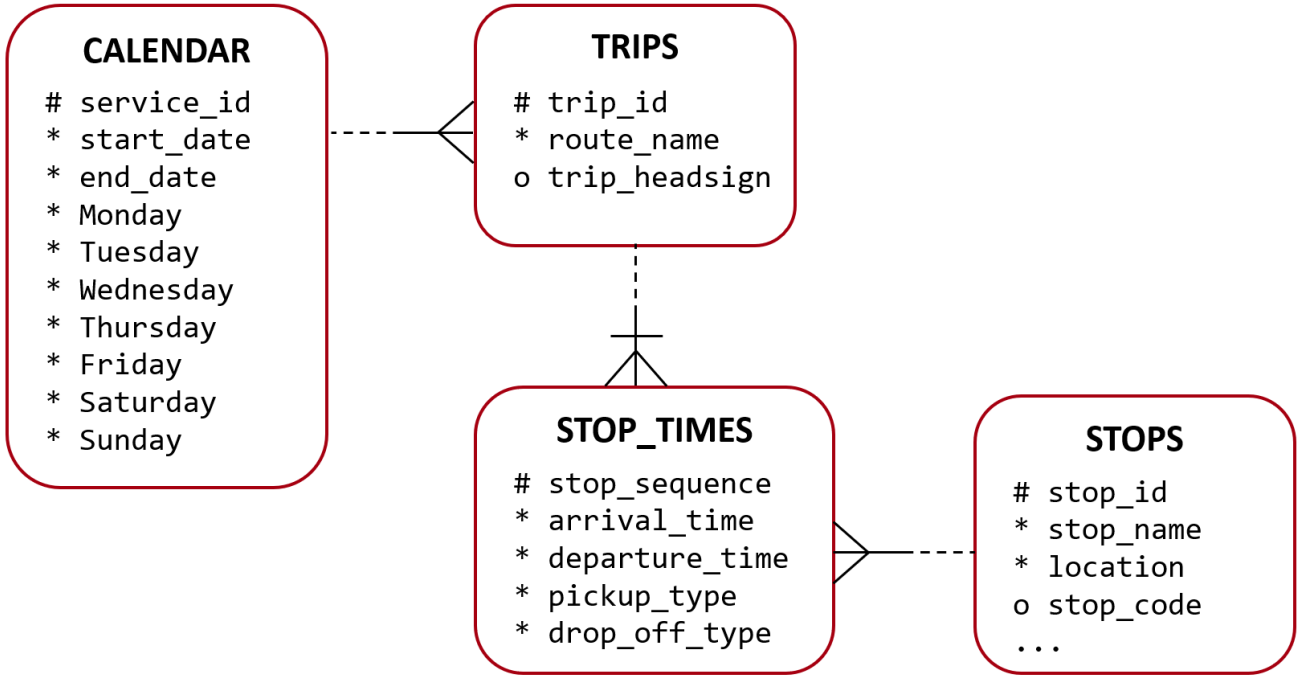


Demo – Data Model



- Persistent external tables
- A procedure for reload





- The "problem" with persistence

```
$ egrep '(ERROR|ORA-)' install.log
ERROR at line 1:
ORA-00942: table or view does not exist
ERROR at line 1:
ORA-00942: table or view does not exist
ERROR at line 1:
ORA-00942: table or view does not exist
```

```
create or replace package gtfs as
  procedure reload;
end gtfs;
```

```
 procedure reload is
  begin
    delete stop_times;
    delete stops;
    delete trips;
    delete calendar;

    load_calendar;
    load_trips;
    load_stops;
    load_stop_times;

    commit;
end reload;
```

```
 procedure reload is
  begin
    delete stop_times;
    delete stops;
    delete trips;
    delete calendar;

    load_calendar;
    load_trips;
    load_stops;
    load_stop_times;

    commit;
end reload;
```

```
 procedure load_calendar is
  begin
    insert into calendar
      (service_id,
       monday,
       tuesday,
       wednesday,
       thursday,
       friday,
       saturday,
       sunday,
       start_date,
       end_date)
      select service_id,
             monday,
             tuesday,
             wednesday,
             thursday,
             friday,
             saturday,
             sunday,
             to_date(start_date, 'yyyymmdd'),
             to_date(end_date, 'yyyymmdd')
      from   calendar_x;
  end load_calendar;
```

What if it's huge?

```
procedure load_calendar is
 begin
   insert into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x;
 end load_calendar;
```

```
procedure load_calendar is
 begin
   insert /*+ APPEND */ into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x;
 end load_calendar;
```

What if it's huge?

```
procedure load_calendar is
 begin
   insert into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x;
 end load_calendar;
```

```
procedure load_calendar is
 begin
   insert /*+ APPEND */ into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x;
 end load_calendar;

select count(*) from calendar
              *
ERROR at line 1:
ORA-12838: cannot read/modify an object
after modifying it in parallel
```

What if it's multiple?

```
procedure load_calendar(
  p_year int default null) is
 begin
   insert /*+ APPEND */ into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x
   where  start_date
     like to_char(p_year)||'%';
 end load_calendar;
```

```
procedure reload is
 begin
   for yr in 2020 .. 2023
   loop
     load_calendar(p_year=>yr);
   end loop;
   load_trips;
   load_stops;
   load_stop_times;
 end reload;
```

What if it's multiple?

```
procedure load_calendar(
  p_year int default null) is
 begin
   insert /*+ APPEND */ into calendar
     (service_id,
      ...
      start_date,
      end_date)
   select
     (service_id,
      ...
      start_date,
      end_date)
   from   calendar_x
   where  start_date
     like to_char(p_year)||'%';
 end load_calendar;
```

```
 procedure reload is
  begin
    for yr in 2020 .. 2023
    loop
      load_calendar(p_year=>yr);
    end loop;
    load_trips;
    load_stops;
    load_stop_times;
  end reload;

SQL> exec gtfs.reload
*
ERROR at line 1:
ORA-12838: cannot read/modify an object
after modifying it in parallel
ORA-06512: at "DEMO.GTFS", line 7
ORA-06512: at "DEMO.GTFS", line 4
ORA-06512: at line 1
```


```
 procedure load_calendar is
  begin
    insert /*+ APPEND */ into calendar
      (service_id,
       ...
       start_date,
       end_date)
    select
      (service_id,
       ...
       start_date,
       end_date)
    from   calendar_x;
  end load_calendar;

select count(*) c from calendar

         C
---------
     12860
```

```
 procedure reload is
  begin
    for yr in 2020 .. 2023
    loop
      load_calendar(p_year=>yr);
    end loop;
    load_trips;
    load_stops;
    load_stop_times;
  end reload;

SQL> exec gtfs.reload

PL/SQL procedure completed successfully.
```

```
 procedure reload is
  begin
    delete stop_times;
    delete stops;
    delete trips;
    delete calendar;

    load_calendar;
    load_trips;
    load_stops;
    load_stop_times;

    commit;
end reload;
```

```
procedure reload is
 begin
   delete stop_times;
   delete stops;
   delete trips;
   delete calendar;

   load_calendar;
   load_trips;
   load_stops;
   load_stop_times;
end
```

trips.txt

```
route_id,service_id,trip_id,trip_headsign,block_id,shape_id,wheelchair_accessible,trip_direction_name
45840,0,VJ71d9dd27f0154e5439851cc2759b97e046091774,"Belgravia",,,0,""
45840,0,VJb901a98e988dc5897ab4eae7aa8643e42b42db55,"Belgravia",,,0,""
45840,0,VJ71d32da0c45a1f74619b1d281b190b047f61b771,"Belgravia",,,0,""
45840,0,VJ753120258014a76706f0f5ffe3876d5f0b3fc428,"Belgravia",,,0,""
45840,0,VJ3c313f9bd790ed6e0fa93098ed3fd4db91905c63,"Belgravia",,,0,""
45840,0,VJ8db6cf13fbce044edf4b6cde1777e6899c64ef81,"Luton Airport",,,0,""
45840,0,VJ43908c0f672d3e968b475e91a90a740414ae6737,"Luton Airport",,,0,""
45840,0,VJ70a8eb954b75178a42f999242a5e2279b792f3cc,"Luton Airport",,,0,""
45840,0,VJ3162628884a261d6e3d690f9268297f753d04bdb,"Luton Airport",,,0,""
31485,1,VJc195461cbdb9b039452a2d10463bf2790022f9bc,"Victoria",,,0,""
31485,2,VJ1dd39883f73952b54b1f2a7fd434a84194b64ac2,"Hull",,,0,""
```

```
procedure reload is
 begin
   delete stop_times;
   delete stops;
   delete trips;
   delete calendar;

   load_calendar;
   load_trips;
   load_stops;
   load_stop_times;
end
```

trips.txt

```
route_id,service_id,trip_id,trip_headsign,block_id,shape_id,wheelchair_accessible,trip_direction_name
45840,0,VJ71d9dd27f0154e5439851cc2759b97e046091774,"Belgravia",,,0,""
45840,0,VJb901a98e988dc5897ab4eae7aa8643e42b42db55,"Belgravia",,,0,""
45840,0,VJ71d32da0c45a1f74619b1d281b190b047f61b771,"Belgravia",,,0,""
45840,0,VJ753120258014a76706f0f5ffe3876d5f0b3fc428,"Belgravia",,,0,""
45840,0,VJ3c313f9bd790ed6e0fa93098ed3fd4db91905c63,"Belgravia",,,0,""
45840,0,VJ8db6cf13fbce044edf4b6cde1777e6899c64ef81,"Luton Airport",,,0,""
45840,0,VJ43908c0f672d3e968b475e91a90a740414ae6737,"Luton Airport",,,0,""
45840,0,VJ70a8eb954b75178a42f999242a5e2279b792f3cc,"Luton Airport",,,0,""
45840,0,VJ3162628884a261d6e3d690f9268297f753d04bdb,"Luton Airport",,,0,""
31485,1,VJc195461cbdb9b039452a2d10463bf2790022f9bc,"Victoria",,,0,""
31485,2,VJ1dd39883f73952b54b1f2a7fd434a84194b64ac2,"Hull",,,0,""
```

```
create sequence trips_seq;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar(service_id),
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
create sequence trips_seq;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar(service_id),
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
create sequence trips_seq;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar(service_id),
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
create sequence trips_seq;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar(service_id),
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
create sequence trips_seq session;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
 procedure reload is
  begin
     delete stop_times;
     delete stops;
     delete trips;
     delete calendar;

     load_calendar;
     load_trips;
     load_stops;
     load_stop_times;

     commit;
end reload;
```

```
create sequence trips_seq session;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

12.1

```
 procedure reload is
  begin
     delete stop_times;
     delete stops;
     delete trips;
     delete calendar;

     load_calendar;
     load_trips;
     load_stops;
     load_stop_times;

     commit;
end reload;
```

```
create sequence trips_seq session;

create table trips (
  trip_id
    integer
    default trips_seq.nextval
    not null
    constraint trips_pk primary key,
  route_name
    varchar2(100)
    not null,
  service_id
    not null
    constraint trips_fk_calendar references calendar(service_id),
  trip_headsign
    varchar2(100),
  trip_x_id
    varchar2(100)
    invisible
    not null
    constraint trips_uk unique
);
```

```
 procedure reload is
  begin
    delete stop_times;
    delete stops;
    delete trips;
    delete calendar;

    load_calendar;
    load_trips;
    load_stops;
    load_stop_times;

    commit;
end reload;
```

```
procedure load_trips is
 begin
   insert into trips
     (trip_x_id,
      route_name,
      service_id,
      trip_headsign)
     select t.trip_id,
            r.route_short_name,
            t.service_id,
            t.trip_headsign
     from   routes_x r,
            trips_x  t,
            calendar c
     where  t.route_id = r.route_id
     and    c.service_id = t.service_id;
 end load_trips;
```

```
procedure reload is
 begin
   delete stop_times;
   delete stops;
   delete trips;
   delete calendar;

   load_calendar;
   load_trips;
   load_stops;
   load_stop_times;

   commit;
end reload;
```

```
procedure load_stops is
 begin
   insert into stops
     (stop_id,
      stop_code,
      stop_name,
      location)
     select stop_id,
            stop_code,
            replace(stop_name, ' Underground Station'),
            mdsys.sdo_geometry(2001, -- 2 dimensional point
                               8307, -- lat/long coord. system
                               mdsys.sdo_point_type(stop_lon,
                                                    stop_lat,
                                                    null),
                               null, -- n/a for point type
                               null) -- n/a for point type
     from   stops_x;
 end load_stops;
```

```
procedure reload is
 begin
   delete stop_times;
   delete stops;
   delete trips;
   delete calendar;

   load_calendar;
   load_trips;
   load_stops;
   load_stop_times;

   commit;
end reload;
```

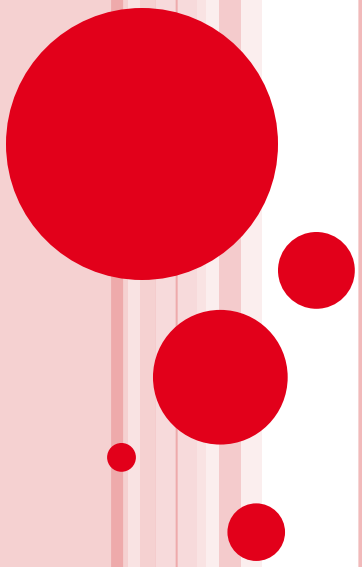
```
procedure load_stops is
 begin
   insert into stops
     (stop_id,
      stop_code,
      stop_name,
      location)
     select stop_id,
            stop_code,
            replace(stop_name, ' Underground Station'),
            mdsys.sdo_geometry(2001, -- 2 dimensional point
                               8307, -- lat/long coord. system
                               mdsys.sdo_point_type(stop_lon,
                                                    stop_lat,
                                                    null),
                               null, -- n/a for point type
                               null) -- n/a for point type
     from   stops_x;
 end load_stops;
```

```
 procedure reload is
  begin
    delete stop_times;
    delete stops;
    delete trips;
    delete calendar;

    load_calendar;
    load_trips;
    load_stops;
    load_stop_times;

    commit;
end reload;
```

```
procedure load_stop_times is
 begin
   insert into stop_times
     (trip_id,
      stop_sequence,
      arrival_time,
      departure_time,
      stop_id,
      pickup_type,
      drop_off_type)
     select t.trip_id,
            st.stop_sequence,
            st.arrival_time,
            st.departure_time,
            st.stop_id,
            st.pickup_type,
            st.drop_off_type
     from   stop_times_x st,
            trips        t
     where  st.trip_id = t.trip_x_id;
 end load_stop_times;
```

External Tables

- ✓ Ad-Hoc Queries
- ✓ Ongoing Loading
- ➢ **Runtime Adjustments**



Overriding Parameters in a Query

- Overridable Clauses
  - DEFAULT DIRECTORY
  - LOCATION
  - ACCESS PARAMETERS
  - REJECT LIMIT

The CALENDAR Table

SERVICE_ID	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	START_DATE	END_DATE
0	1	1	1	1	1	1	1	07/10/2024	16/04/2025
1	1	1	1	1	1	1	1	07/10/2024	16/04/2025
2	0	0	0	0	1	0	0	11/10/2024	11/04/2025
3	1	1	1	1	0	1	1	07/10/2024	16/04/2025
4	1	1	1	1	1	1	1	07/10/2024	27/03/2025
5	0	0	0	0	1	1	1	07/10/2024	23/03/2025
6	1	1	1	1	1	1	1	07/10/2024	16/04/2025
7	0	0	0	0	1	0	0	07/10/2024	16/04/2025
8	1	1	1	1	0	1	1	07/10/2024	16/04/2025
9	1	1	1	1	1	1	1	07/10/2024	16/04/2025
10	1	1	1	1	1	1	1	07/10/2024	16/04/2025
11	1	1	1	1	1	1	1	07/10/2024	27/03/2025
12	0	0	0	0	1	1	1	07/10/2024	23/03/2025
13	1	1	1	1	1	0	0	07/10/2024	16/04/2025
14	0	0	0	0	0	1	1	07/10/2024	16/04/2025
15	1	1	1	1	1	0	0	07/10/2024	16/04/2025
16	0	0	0	0	0	1	1	07/10/2024	16/04/2025
17	0	0	1	0	0	0	0	07/10/2024	16/04/2025
18	0	0	0	1	0	0	0	07/10/2024	16/04/2025
19	1	1	1	1	1	1	1	07/10/2024	16/04/2025

The CALENDAR Table

SERVICE_ID	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	START_DATE	END_DATE
0	1	1	1	1	1	1	1	07/10/2024	16/04/2025
1	1	1	1	1	1	1	1	07/10/2024	16/04/2025
2	0	0	0	0	1	0	0	11/10/2024	11/04/2025
3	1	1	1	1	0	1	1	07/10/2024	16/04/2025
4	1	1	1	1	1	1	1	07/10/2024	27/03/2025
5	0	0	0	0	1	1	1	07/10/2024	23/03/2025
6	1	1	1	1	1	1	1	07/10/2024	16/04/2025
7	0	0	0	0	1	0	0	07/10/2024	16/04/2025
8	1	1	1	1	0	1	1	07/10/2024	16/04/2025
9	1	1	1	1	1	1	1	07/10/2024	16/04/2025
10	1	1	1	1	1	1	1	07/10/2024	16/04/2025
11	1	1	1	1	1	1	1	07/10/2024	27/03/2025
12	0	0	0	0	1	1	1	07/10/2024	23/03/2025
13	1	1	1	1	1	0	0	07/10/2024	16/04/2025
14	0	0	0	0	0	1	1	07/10/2024	16/04/2025
15	1	1	1	1	1	0	0	07/10/2024	16/04/2025
16	0	0	0	0	0	1	1	07/10/2024	16/04/2025
17	0	0	1	0	0	0	0	07/10/2024	16/04/2025
18	0	0	0	1	0	0	0	07/10/2024	16/04/2025
19	1	1	1	1	1	1	1	07/10/2024	16/04/2025

The CALENDAR Table

SERVICE_ID	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	START_DATE	END_DATE
0	1	1	1	1	1	1	1	07/10/2024	16/04/2025
1	1	1	1	1	1	1	1	07/10/2024	16/04/2025
2	0	0	0	0	1	0	0	11/10/2024	11/04/2025
3	1	1	1	1	0	1	1	07/10/2024	16/04/2025
4	1	1	1	1	1	1	1	07/10/2024	27/03/2025
5	0	0	0	0	1	1	1	07/10/2024	23/03/2025
6	1	1	1	1	1	1	1	07/10/2024	16/04/2025
7	0	0	0	0	1	0	0	07/10/2024	16/04/2025
8	1	1	1	1	0	1	1	07/10/2024	16/04/2025
9	1	1	1	1	1	1	1	07/10/2024	16/04/2025
10	1	1	1	1	1	1	1	07/10/2024	16/04/2025
11	1	1	1	1	1	1	1	07/10/2024	27/03/2025
12	0	0	0	0	1	1	1	07/10/2024	23/03/2025
13	1	1	1	1	1	0	0	07/10/2024	16/04/2025
14	0	0	0	0	0	1	1	07/10/2024	16/04/2025
15	1	1	1	1	1	0	0	07/10/2024	16/04/2025
16	0	0	0	0	0	1	1	07/10/2024	16/04/2025
17	0	0	1	0	0	0	0	07/10/2024	16/04/2025
18	0	0	0	1	0	0	0	07/10/2024	16/04/2025
19	1	1	1	1	1	1	1	07/10/2024	16/04/2025

The CALENDAR Table

SERVICE_ID	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	START_DATE	END_DATE
0	1	1	1	1	1	1	1	07/10/2024	16/04/2025
1	1	1	1	1	1	1	1	07/10/2024	16/04/2025
2	0	0	0	0	1	0	0	11/10/2024	11/04/2025
3	1	1	1	1	0	1	1	07/10/2024	16/04/2025
4	1	1	1	1	1	1	1	07/10/2024	27/03/2025
5	0	0	0	0	1	1	1	07/10/2024	23/03/2025
6	1	1	1	1	1	1	1	07/10/2024	16/04/2025
7	0	0	0	0	1	0	0	07/10/2024	16/04/2025
8	1	1	1	1	0	1	1	07/10/2024	16/04/2025
9	1	1	1	1	1	1	1	07/10/2024	16/04/2025
10	1	1	1	1	1	1	1	07/10/2024	16/04/2025
11	1	1	1	1	1	1	1	07/10/2024	27/03/2025
12	0	0	0	0	1	1	1	07/10/2024	23/03/2025
13	1	1	1	1	1	0	0	07/10/2024	16/04/2025
14	0	0	0	0	0	1	1	07/10/2024	16/04/2025
15	1	1	1	1	1	0	0	07/10/2024	16/04/2025
16	0	0	0	0	0	1	1	07/10/2024	16/04/2025
17	0	0	1	0	0	0	0	07/10/2024	16/04/2025
18	0	0	0	1	0	0	0	07/10/2024	16/04/2025
19	1	1	1	1	1	1	1	07/10/2024	16/04/2025

Can we do better than 1s and 0s?

```
SQL> select * from calendar;

SERVICE_ID  MON  TUE  WED  THU  FRI  SAT  SUN START_DAT END_DATE
---------- ---- ---- ---- ---- ---- ---- ---- --------- ---------
     58736    0    1    0    1    1    1    0 11-SEP-22 17-NOV-22
       947    1    1    0    1    0    1    0 10-OCT-22 01-DEC-22
      1436    1    1    0    1    0    0    0 19-OCT-22 24-NOV-22
      ...
```

Can we do better than 1s and 0s?


```
create table calendar23 (
    service_id number(4) not null,
    monday     boolean   not null,
    tuesday    boolean   not null,
    wednesday  boolean   not null,
    thursday   boolean   not null,
    friday     boolean   not null,
    saturday   boolean   not null,
    sunday     boolean   not null,
    start_date date      not null,
    end_date   date      not null
);
```

Can we do better than 1s and 0s?

```
begin
  dbms_redefinition.start_redef_table(
     user, 'calendar', 'calendar23',
     col_mapping=>
       'service_id             service_id,
        to_boolean(monday)     monday,
        to_boolean(tuesday)    tuesday,
        to_boolean(wednesday)  wednesday,
        to_boolean(thursday)   thursday,
        to_boolean(friday)     friday,
        to_boolean(saturday)   saturday,
        to_boolean(sunday)     sunday,
        start_date             start_date,
        end_date               end_date'
     );
end;
```

Can we do better than 1s and 0s?

```
alter table calendar23 add
  constraint calendar23_pk primary key (service_id);

begin
  dbms_redefinition.finish_redef_table(
     user, 'calendar', 'calendar23');
end;
```

Can we do better than 1s and 0s?

```
SQL> desc calendar

 Name                          Null?    Type
 ----------------------------- -------- -----------
 SERVICE_ID                    NOT NULL NUMBER(4)
 MONDAY                        NOT NULL BOOLEAN
 TUESDAY                       NOT NULL BOOLEAN
 WEDNESDAY                     NOT NULL BOOLEAN
 THURSDAY                      NOT NULL BOOLEAN
 FRIDAY                        NOT NULL BOOLEAN
 SATURDAY                      NOT NULL BOOLEAN
 SUNDAY                        NOT NULL BOOLEAN
 START_DATE                    NOT NULL DATE
 END_DATE                      NOT NULL DATE
```

```
select c.*
from   calendar c
where  trunc(sysdate) between c.start_date and c.end_date
and    decode(to_char(sysdate, 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1;
```

```
select c.*
from   calendar c
where  trunc(sysdate) between c.start_date and c.end_date
and    decode(to_char(sysdate, 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1;
```

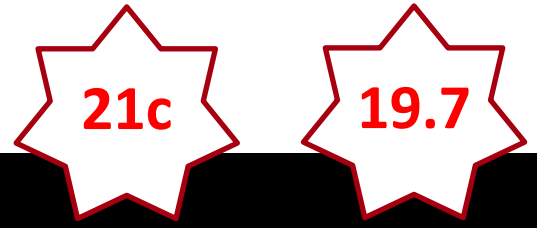
```
select c.*
from   calendar c
where  trunc(sysdate) between c.start_date and c.end_date
and    decode(to_char(sysdate, 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1;
```

```
select c.*
from   calendar c
where  trunc(sysdate+1) between c.start_date and c.end_date
and    decode(to_char(sysdate+1, 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1;
```

```
select c.*
from   calendar c
where  trunc(date'2024-08-29') between c.start_date and c.end_date
and    decode(to_char(date'2024-08-29', 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1;
```

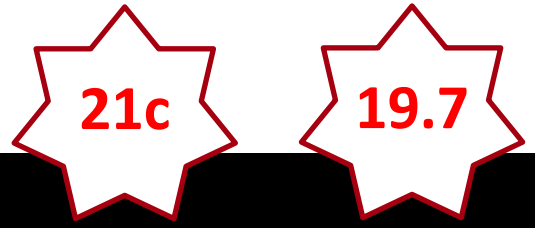
Parameterized Views?

Table SQL Macro – Parameterized Views




```
create or replace package body gtfs_sm as

  function services(i_date in date default sysdate)
    return varchar2 sql_macro(table) is
  begin
    return q'<
select c.service_id
from   calendar c
where  trunc(i_date) between c.start_date and c.end_date
and    decode(to_char(i_date, 'fmDay', 'nls_date_language=english'),
              'Sunday',   c.sunday,
              'Monday',   c.monday,
              'Tuesday',  c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',   c.friday,
              'Saturday', c.saturday) = 1
>';
  end services;
end gtfs_sm;
```

Table SQL Macro – Parameterized Views




```
create or replace package body gtfs_sm as

  function services(i_date in date default sysdate)
    return varchar2 sql_macro(table) is
  begin
    return q'<
select c.service_id
from   calendar c
where  trunc(i_date) between c.start_date and c.end_date
and    decode(to_char(i_date, 'fmDay', 'nls_date_language=english'),
              'Sunday',    c.sunday,
              'Monday',    c.monday,
              'Tuesday',   c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',    c.friday,
              'Saturday', c.saturday) = 1
>';
  end services;
end gtfs_sm;
```

Table SQL Macro – Parameterized Views

21c

19.7

```
create or replace package body gtfs_sm as

  function services(i_date in date default sysdate)
    return varchar2 sql_macro(table) is
  begin
    return q'<
select c.service_id
from   calendar c
where  trunc(i_date) between c.start_date and c.end_date
and    decode(to_char(i_date, 'fmDay', 'nls_date_language=english'),
              'Sunday',    c.sunday,
              'Monday',    c.monday,
              'Tuesday',   c.tuesday,
              'Wednesday',c.wednesday,
              'Thursday', c.thursday,
              'Friday',    c.friday,
              'Saturday', c.saturday) = 1
  end services;
end gtfs_sm;
```

@sm1


```
declare
  v_in clob := q'<select t.route_name,count(*) number_of_trips
                 from gtfs_sm.services(sysdate) s, trips t
                 where  t.service_id = s.service_id
                 group  by t.route_name
                 order  by number_of_trips>';
  v_out clob;
begin
  dbms_utility.expand_sql_text(input_sql_text => v_in,
                               output_sql_text => v_out);
  dbms_output.put_line(v_out);
end;
/
```

```
select t.route_name,count(*) number_of_trips
from gtfs_sm.services(sysdate) s, trips t
where  t.service_id = s.service_id
group  by t.route_name
order  by number_of_trips
```

```
SELECT "A1"."ROUTE_NAME" "ROUTE_NAME",
       COUNT(*) "NUMBER_OF_TRIPS"
FROM   (SELECT "A3"."SERVICE_ID" "SERVICE_ID"
        FROM   (SELECT "A4"."SERVICE_ID" "SERVICE_ID"
                FROM   "DEMO_WLWM"."CALENDAR" "A4"
                WHERE  TRUNC(SYSDATE) >= "A4"."START_DATE"
                AND    TRUNC(SYSDATE) <= "A4"."END_DATE"
                AND    DECODE(TO_CHAR(SYSDATE, 'fmDay', 'nls_date_language=english'),
                              'Sunday',"A4"."SUNDAY",
                              'Monday',"A4"."MONDAY",
                              'Tuesday',"A4"."TUESDAY",
                              'Wednesday',"A4"."WEDNESDAY",
                              'Thursday',"A4"."THURSDAY",
                              'Friday',"A4"."FRIDAY",
                              'Saturday',"A4"."SATURDAY") = 1) "A3") "A2",
       "DEMO_WLWM"."TRIPS" "A1"
WHERE  "A1"."SERVICE_ID" = "A2"."SERVICE_ID"
GROUP  BY "A1"."ROUTE_NAME"
ORDER  BY COUNT(*)
```

```
21c> select d.day,
  2         s.service_id
  3  from   (select trunc(sysdate)+level-1 day from dual connect by level <= 3) d,
  4         gtfs_sm.services(d.day) s
  5  order by s.service_id, d.day;
```

```
21c> select d.day,
  2         s.service_id
  3  from   (select trunc(sysdate)+level-1 day from dual connect by level <= 3) d,
  4         gtfs_sm.services(d.day) s
  5  order by s.service_id, d.day;
```

```
21c> select d.day,
  2         s.service_id
  3  from   (select trunc(sysdate)+level-1 day from dual connect by level <= 3) d,
  4         gtfs_sm.services(d.day) s
  5  order by s.service_id, d.day;

DAY        SERVICE_ID
---------- ----------
2024-10-24          0
2024-10-25          0
2024-10-26          0
2024-10-24          1
2024-10-25          1
2024-10-26          1
2024-10-25          2
2024-10-24          3
2024-10-26          3
2024-10-24          4
...
```

The obsession with DUAL ☺

```
23c> select d.day,
  2         s.service_id
  3  from   (select trunc(sysdate)+level-1 day                connect by level <= 3) d,
  4         gtfs_sm.services(d.day) s
  5  order by s.service_id, d.day;

DAY         SERVICE_ID
----------- ----------
2024-10-24           0
2024-10-25           0
2024-10-26           0
2024-10-24           1
2024-10-25           1
2024-10-26           1
2024-10-25           2
2024-10-24           3
2024-10-26           3
2024-10-24           4
...
```

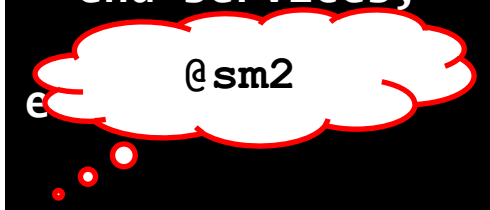
23ai

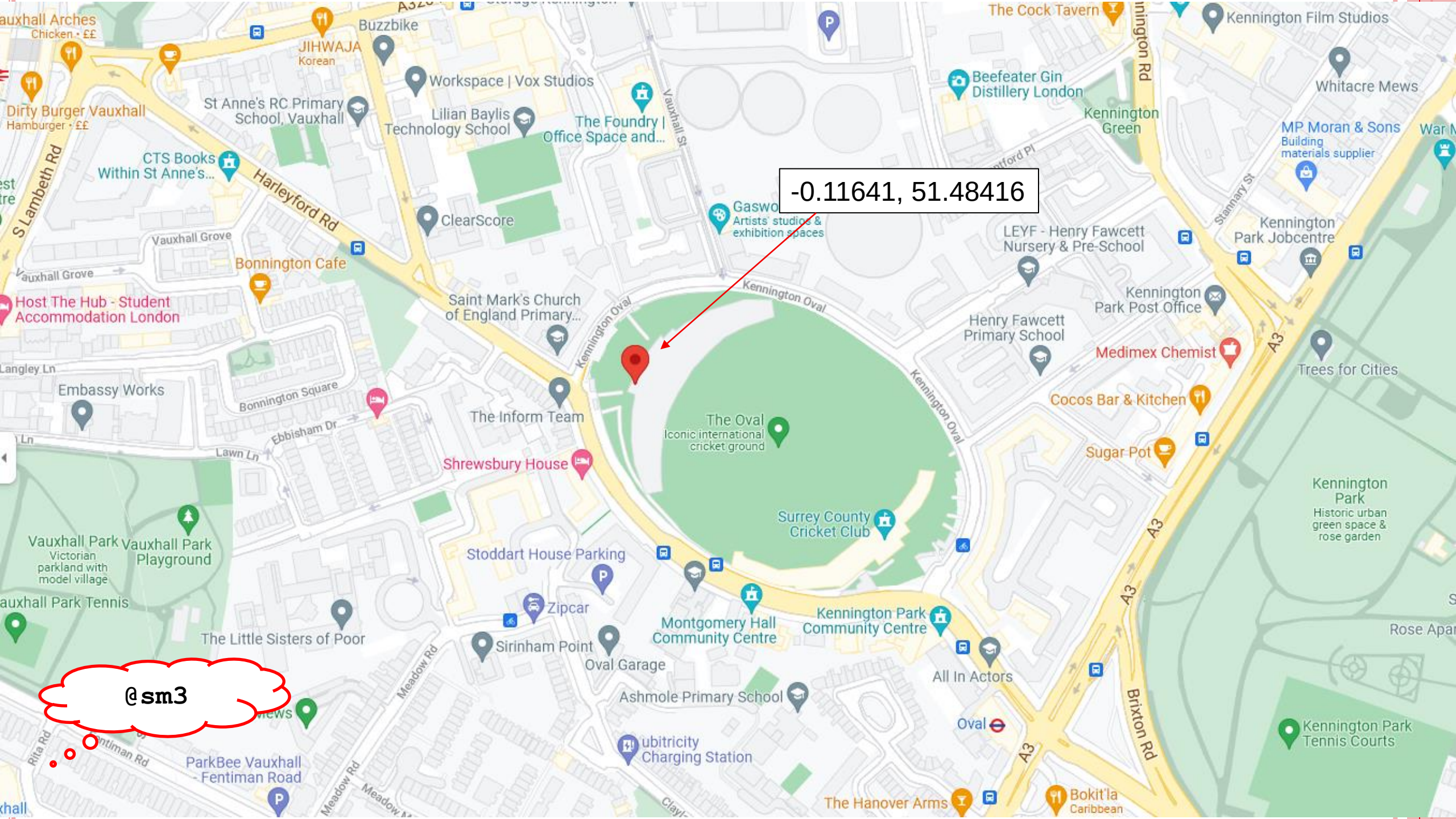
```
21c> select d.day,
  2         s.service_id
  3  from   (select trunc(sysdate)+level-1 day from dual connect by level <= 3) d,
  4         gtfs_sm.services(d.day) s
  5  order by s.service_id, d.day;

DAY         SERVICE_ID
----------- ----------
2024-10-24           0
2024-10-25           0
2024-10-26           0
2024-10-24           1
2024-10-25           1
2024-10-26           1
2024-10-25           2
2024-10-24           3
2024-10-26           3
2024-10-24           4
...
```

```
create or replace package gtfs_sm as

  function services(i_date in date default sysdate) return varchar2 sql_macro(table);

  function services
  (
    i_from_date in date,
    i_days      in integer
  ) return varchar2 sql_macro(table);

end gtfs_sm;
```

```
create or replace package body gtfs_sm as
  function services(i_date in date default sysdate) return varchar2 sql_macro(table) is
  ...
  end services;

  function services
  (
    i_from_date in date,
    i_days      in integer
  ) return varchar2 sql_macro(table) is
  begin
    return '
select day, service_id
from (select trunc(i_from_date)+level-1 day from dual connect by level<=i_days),
     gtfs_sm.services(day)';
  end services;

end gtfs_sm;
```

```
create or replace package body gtfs_sm as
  function services(i_date in date default sysdate) return varchar2 sql_macro(table) is
  ...
  end services;

  function services
  (
    i_from_date in date,
    i_days      in integer
  ) return varchar2 sql_macro(table) is
  begin
    return '
select day, service_id
from (select trunc(i_from_date)+level-1 day from dual connect by level<=i_days),
     gtfs_sm.services(day)';
  end services;
e
```




-0.11641, 51.48416
@sm3
The Oval
Iconic international cricket ground
Kennington Oval
Harleyford Rd
S Lambeth Rd
Vauxhall Grove
Bonnington Square
Ebbisham Dr
Lawn Ln
Langley Ln
Meadow Rd
Rita Rd
Fentiman Rd
Vauxhall St
Stannary St
Brixton Rd
A3
Vauxhall Arches
Chicken · ££
Dirty Burger Vauxhall
Hamburger · ££
JIHWAJA
Korean
Buzzbike
Workspace | Vox Studios
St Anne's RC Primary School, Vauxhall
Lilian Baylis Technology School
The Foundry | Office Space and...
CTS Books Within St Anne's...
ClearScore
Bonnington Cafe
Host The Hub - Student Accommodation London
Saint Mark's Church of England Primary...
Embassy Works
The Inform Team
Shrewsbury House
Vauxhall Park
Victorian parkland with model village
Vauxhall Park Playground
Vauxhall Park Tennis
The Little Sisters of Poor
Stoddart House Parking
Zipcar
Sirinham Point
Oval Garage
Montgomery Hall Community Centre
Surrey County Cricket Club
Kennington Park Community Centre
All In Actors
Ashmole Primary School
ubitricity Charging Station
Oval
ParkBee Vauxhall - Fentiman Road
The Hanover Arms
Bokit'la
Caribbean
The Cock Tavern
Kennington Film Studios
Beefeater Gin Distillery London
Whitacre Mews
Kennington Green
MP Moran & Sons
Building materials supplier
Gaswo
Artists' studios & exhibition spaces
LEYF - Henry Fawcett Nursery & Pre-School
Kennington Park Jobcentre
Kennington Park Post Office
Henry Fawcett Primary School
Medimex Chemist
Trees for Cities
Cocos Bar & Kitchen
Sugar Pot
Kennington Park
Historic urban green space & rose garden
Kennington Park Tennis Courts

```
create or replace function nearby_stops
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number
) return varchar2 sql_macro(table) as
begin
  return '
    select *
    from stops s
    where sdo_within_distance(
          s.location,
          sdo_geometry(2001, 8307,
                       sdo_point_type(i_longitude, i_latitude, null), null, null),
                       ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
end nearby_stops;
/
```

```
create or replace function nearby_stops
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number
) return varchar2 sql_macro(table) as
begin
  return '
    select *
    from stops s
    where sdo_within_distance(
          s.location,
          sdo_geometry(2001, 8307,
                       sdo_point_type(i_longitude, i_latitude, null), null, null),
                       ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
end nearby_stops;
/
```

```
create or replace function nearby_stops
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number
) return varchar2 sql_macro(table) as
begin
  return '
    select *
    from stops s
    where sdo_within_distance(
          s.location,
          sdo_geometry(2001, 8307,
                       sdo_point_type(i_longitude, i_latitude, null), null, null),
                       ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
end nearby_stops;
/
```

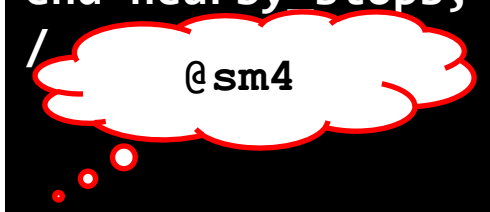


Table SQL Macro – Polymorphic Views

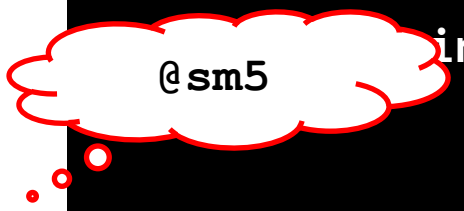
```
create or replace function nearby_points
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number,
  i_table     in dbms_tf.table_t,
  i_column    in dbms_tf.columns_t
) return varchar2 sql_macro(table) as
begin
  return 'select * from i_table s
  where sdo_within_distance(
                  s.' || i_column(1) || ',
                  sdo_geometry(2001, 8307,
                               sdo_point_type(i_longitude, i_latitude, null),
                               null, null),
                  ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
end nearby_points;
/
```

Table SQL Macro – Polymorphic Views

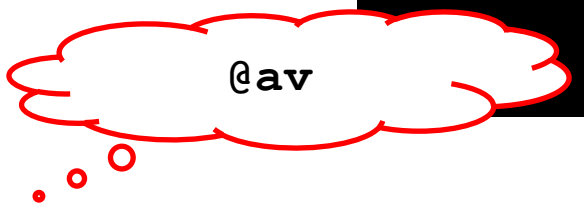
```
create or replace function nearby_points
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number,
  i_table     in dbms_tf.table_t,
  i_column    in dbms_tf.columns_t
) return varchar2 sql_macro(table) as
begin
  return 'select * from i_table s
  where sdo_within_distance(
                  s.' || i_column(1) || ',
                  sdo_geometry(2001, 8307,
                               sdo_point_type(i_longitude, i_latitude, null),
                               null, null),
                  ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
end nearby_points;
/
```

Table SQL Macro – Polymorphic Views

```
create or replace function nearby_points
(
  i_longitude in number,
  i_latitude  in number,
  i_radius    in number,
  i_table     in dbms_tf.table_t,
  i_column    in dbms_tf.columns_t
) return varchar2 sql_macro(table) as
begin
  return 'select * from i_table s
  where sdo_within_distance(
                  s.' || i_column(1) || ',
                  sdo_geometry(2001, 8307,
                               sdo_point_type(i_longitude, i_latitude, null),
                               null, null),
                  ''distance=' || i_radius || ' unit=meter'') = ''TRUE''';
                  ints;
```



```
select s.stop_id,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips;
```



```
select s.stop_id,
       s.stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips;
```

```
select s.stop_id,
       s.stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips;

ERROR at line 2:
ORA-00979: not a GROUP BY expression
```

When error messages don't give the message

```
select s.stop_id,
       s.stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips;

ERROR at line 2:
ORA-00979: not a GROUP BY expression
```

When error messages don't give the message

```
select s.stop_id,
       s.stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips;

ERROR at line 2:
ORA-00979: "STOP_NAME": must appear in the GROUP BY clause or be
used in an aggregate function
```

Option 1

```
select s.stop_id,
       s.stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id,
          s.stop_name
order  by number_of_trips
```

Option 2

```
select s.stop_id,
       min(s.stop_name) stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips
```

Option 3 – ANY_VALUE


```
select s.stop_id,
       any_value(s.stop_name) stop_name,
       count(*) number_of_trips,
       sum(st.pickup_type) no_pickups,
       sum(st.drop_off_type) no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       stop_times st
where  st.stop_id = s.stop_id
group  by s.stop_id
order  by number_of_trips
```

Option 4

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       (select st.stop_id,
               count(*) number_of_trips,
               sum(st.pickup_type) no_pickups,
               sum(st.drop_off_type) no_drop_offs
        from   stop_times st
        group  by st.stop_id) st1
where  st1.stop_id = s.stop_id
order  by st1.number_of_trips;
```

Option 4

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       (select st.stop_id,
               count(*) number_of_trips,
               sum(st.pickup_type) no_pickups,
               sum(st.drop_off_type) no_drop_offs
        from   stop_times st
        group  by st.stop_id) st1
where  st1.stop_id = s.stop_id
order  by st1.number_of_trips;
```

Option 4 - don't forget the join!

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       (select st.stop_id,
               count(*) number_of_trips,
               sum(st.pickup_type) no_pickups,
               sum(st.drop_off_type) no_drop_offs
        from   stop_times st
        group  by st.stop_id) st1
order  by st1.number_of_trips;
```

Option 4 - don't forget the join!

```
-------------------------------------------------------------------------
| Id  | Operation                                    | Name               |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT                             |                    |
|   1 |  SORT ORDER BY                               |                    |
|   2 |   MERGE JOIN CARTESIAN                       |                    |
|   3 |    TABLE ACCESS BY INDEX ROWID               | STOPS              |
|*  4 |     DOMAIN INDEX (SEL: 0.000000 %)           | STOPS_IX_LOCATION  |
|   5 |    BUFFER SORT                               |                    |
|   6 |     VIEW                                     |                    |
|   7 |      HASH GROUP BY                           |                    |
|   8 |       TABLE ACCESS BY INDEX ROWID BATCHED    | STOP_TIMES         |
|   9 |        INDEX FULL SCAN                       | STOP_TIMES_IX_STOP |
-------------------------------------------------------------------------

SQL Analysis Report (identified by operation id/Query Block Name/Object
-------------------------------------------------------------------------
   1 -  SEL$5C160134
           -  The query block has 1 cartesian product which may be
              expensive. Consider adding join conditions or removing the
              disconnected tables or views.
```

Option 5

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
              (select count(*) number_of_trips,
                      sum(st.pickup_type) no_pickups,
                      sum(st.drop_off_type) no_drop_offs
               from   stop_times st
               where  st.stop_id = s.stop_id) st1
order  by st1.number_of_trips;
```

Option 5

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
              (select count(*) number_of_trips,
                      sum(st.pickup_type) no_pickups,
                      sum(st.drop_off_type) no_drop_offs
               from   stop_times st
               where  st.stop_id = s.stop_id) st1
order  by st1.number_of_trips;
```

Option 5

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
              (select count(*) number_of_trips,
                      sum(st.pickup_type) no_pickups,
                      sum(st.drop_off_type) no_drop_offs
               from   stop_times st
               where  st.stop_id = s.stop_id) st1
order  by st1.number_of_trips;

ERROR at line 11:
ORA-00904: "S"."STOP_ID": invalid identifier
```

Option 5 – LATERAL Inline Views

```
select s.stop_id,
       s.stop_name,
       st1.number_of_trips,
       st1.no_pickups,
       st1.no_drop_offs
from   nearby_stops(-0.11641, 51.48416, 1000) s,
       lateral (select count(*) number_of_trips,
                       sum(st.pickup_type) no_pickups,
                       sum(st.drop_off_type) no_drop_offs
                from   stop_times st
                where  st.stop_id = s.stop_id) st1
order  by st1.number_of_trips;
```

12.1

```
select st.*,
       s.stop_name
from   stop_times st,
       stops      s
where  st.trip_id = 171
and    s.stop_id = st.stop_id
order  by st.stop_sequence;
```

TRIP_ID	STOP_SEQUENCE	ARRIVAL_TIME	DEPARTURE_TIME	STOP_ID	PICKUP_TYPE	DROP_OFF_TYPE	STOP_NAME
171	0	11:10:00	11:10:00	9400ZZLUEAC1	0	1	Elephant & Castle
171	1	11:11:00	11:12:00	9400ZZLULBN1	0	0	Lambeth North
171	2	11:13:00	11:14:00	9400ZZLUWLO1	0	0	Waterloo
171	3	11:15:00	11:15:00	9400ZZLUEMB1	0	0	Embankment
171	4	11:16:00	11:16:00	9400ZZLUCHX1	0	0	Charing Cross
171	5	11:18:00	11:18:00	9400ZZLUPCC1	0	0	Piccadilly Circus
171	6	11:20:00	11:20:00	9400ZZLUOXC4	0	0	Oxford Circus
171	7	11:22:00	11:22:00	9400ZZLURGP1	0	0	Regent's Park
171	8	11:24:00	11:24:00	9400ZZLUBST2	0	0	Baker Street
171	9	11:25:00	11:25:00	9400ZZLUMYB1	0	0	Marylebone
171	10	11:26:00	11:26:00	9400ZZLUERB1	0	0	Edgware Road
171	11	11:28:00	11:28:00	9400ZZLUPAC2	0	0	Paddington
171	12	11:30:00	11:30:00	9400ZZLUWKA1	0	0	Warwick Avenue
171	13	11:31:00	11:32:00	9400ZZLUMVL1	0	0	Maida Vale
171	14	11:33:00	11:34:00	9400ZZLUKPK1	0	0	Kilburn Park
171	15	11:36:00	11:36:00	9400ZZLUQPS1	1	0	Queen's Park

```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  ? first_stop_name,
  ? last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```

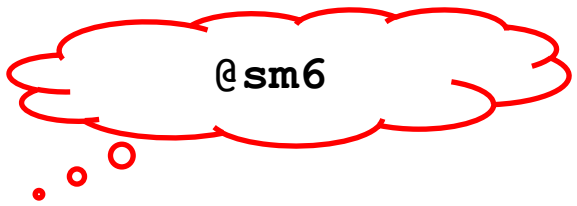
```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  any_value(s.stop_name) keep(dense_rank first order by st.stop_sequence) first_stop_name,
  any_value(s.stop_name) keep(dense_rank last order by st.stop_sequence) last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```

```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  any_value(s.stop_name) keep(dense_rank first order by st.stop_sequence) first_stop_name,
  any_value(s.stop_name) keep(dense_rank last order by st.stop_sequence) last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```

```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  any_value(s.stop_name) keep(dense_rank first order by st.stop_sequence) first_stop_name,
  any_value(s.stop_name) keep(dense_rank last order by st.stop_sequence) last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```

```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  any_value(s.stop_name) keep(dense_rank first order by st.stop_sequence) first_stop_name,
  any_value(s.stop_name) keep(dense_rank last order by st.stop_sequence) last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```

```
create or replace view trip_end_points_v as
select
  st.trip_id,
  min(st.departure_time) trip_start_time,
  max(st.arrival_time) trip_end_time,
  any_value(s.stop_name) keep(dense_rank first order by st.stop_sequence) first_stop_name,
  any_value(s.stop_name) keep(dense_rank last order by st.stop_sequence) last_stop_name
from   stop_times st,
       stops      s
where  s.stop_id = st.stop_id
group  by st.trip_id;
```


@sm6

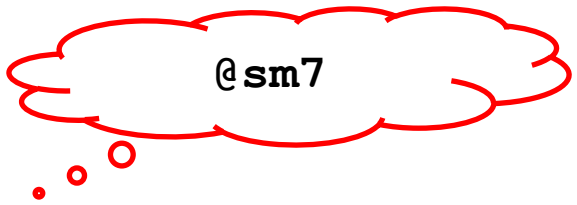
```
create function datetime
(
  i_date in date,
  i_time in varchar2
) return varchar2                    is
begin
  return    trunc(i_date)
            + numtodsinterval(to_number(substr(i_time, 1, 2)), 'hour’)
            + numtodsinterval(to_number(substr(i_time, 4, 2)), 'minute')
            + numtodsinterval(to_number(substr(i_time, 7, 2)), 'second')  ;
end datetime;
/
```

Scalar SQL Macro


```
create function datetime
(
  i_date in date,
  i_time in varchar2
) return varchar2 sql_macro(scalar) is
begin
  return q'<trunc(i_date)
            + numtodsinterval(to_number(substr(i_time, 1, 2)), 'hour’)
            + numtodsinterval(to_number(substr(i_time, 4, 2)), 'minute')
            + numtodsinterval(to_number(substr(i_time, 7, 2)), 'second')>';
end datetime;
/
```

Scalar SQL Macro


```
create function datetime
(
  i_date in date,
  i_time in varchar2
) return varchar2 sql_macro(scalar) is
begin
  return q'<trunc(i_date)
            + numtodsinterval(to_number(substr(i_time, 1, 2)), 'hour’)
            + numtodsinterval(to_number(substr(i_time, 4, 2)), 'minute')
            + numtodsinterval(to_number(substr(i_time, 7, 2)), 'second')>';
end datetime;
/
```



What if you forgot?

```
create or replace function datetime_non_macro
(
  i_date in date,
  i_time in varchar2
) return date is
begin
  return trunc(i_date)
            + numtodsinterval(to_number(substr(i_time, 1, 2)), 'hour')
            + numtodsinterval(to_number(substr(i_time, 4, 2)), 'minute')
            + numtodsinterval(to_number(substr(i_time, 7, 2)), 'second');
end;
/
```

What if you forgot?

```
select
  trip_id,
  datetime_non_macro(sysdate,trip_start_time) true_start,
  datetime_non_macro(sysdate,trip_end_time) true_end
from   trip_end_points_v
where  datetime_non_macro(sysdate,trip_start_time) > sysdate;
```

What if you forgot?


```
SQL> alter session set sql_transpiler = on;

Session altered.

select
  trip_id,
  datetime_non_macro(sysdate,trip_start_time) true_start,
  datetime_non_macro(sysdate,trip_end_time) true_end
from   trip_end_points_v
where  datetime_non_macro(sysdate,trip_start_time) > sysdate;
```

What if you forgot?

```
-----------------------------------------------------
| Id  | Operation          | Name                | Rows  |
-----------------------------------------------------
|   0 | SELECT STATEMENT   |                     |     1 |
|*  1 |  TABLE ACCESS FULL| TRIP_END_POINTS_V |     1 |
-----------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(TRUNC(SYSDATE@!)+NUMTODSINTERVAL(TO_NUMBER(
        SUBSTR("TRIP_START_TIME",1,2)),'hour')+NUMTODSINTERVAL(
        TO_NUMBER(SUBSTR("TRIP_START_TIME",4,2)),'minute')+
        NUMTODSINTERVAL(TO_NUMBER(SUBSTR("TRIP_START_TIME",7,2)),'second')
        >SYSDATE@!)
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

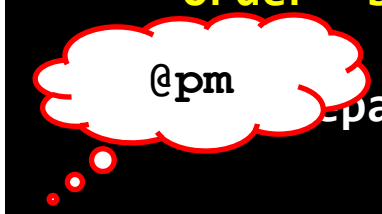
```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```

```
create or replace package body gtfs as
  ...
 procedure get_departures
  (
    i_stop_id   in stops.stop_id%type,
    i_from_date in date default sysdate,
    i_days      in integer default 1,
    o_rc        out sys_refcursor
  ) is
  begin
    open o_rc for
      select t.route_name,
             datetime(s.day, st.departure_time) departure_time
      from   stop_times st,
             trips t,
             gtfs_sm.services(i_from_date, i_days) s
      where  st.stop_id = i_stop_id
      and    t.trip_id = st.trip_id
      and    s.service_id = t.service_id
      order  by t.route_name,
                departure_time;
  end get_departures;
```



```
   (
  select t.route_name,
         datetime(s.day, st.departure_time) departure_time
  from   stop_times st, trips t, gtfs_sm.services(i_from_date, i_days) s
  where  st.stop_id = i_stop_id
  and    t.trip_id = st.trip_id
  and    s.service_id = t.service_id
)
```

```
select route_name, number_of_trips,
       to_char(from_time,'hh24:mi')
         ||nvl2(to_time,
                ' - '||to_char(to_time,'hh24:mi')||' every '||minutes||' minutes',
                null) departures
    (
  select t.route_name,
         datetime(s.day, st.departure_time) departure_time
  from   stop_times st, trips t, gtfs_sm.services(i_from_date, i_days) s
  where  st.stop_id = i_stop_id
  and    t.trip_id = st.trip_id
  and    s.service_id = t.service_id
) match_recognize (
    partition by route_name
    order by departure_time
    measures X.departure_time from_time,
             last(Y.departure_time)  to_time,
             (first(Y.departure_time)-X.departure_time)*24*60 minutes,
             count(*) number_of_trips
    pattern (X Y{2,} | X)
    define
      Y as departure_time-prev(departure_time) = first(Y.departure_time)-X.departure_time
)
order by route_name, from_time;
```

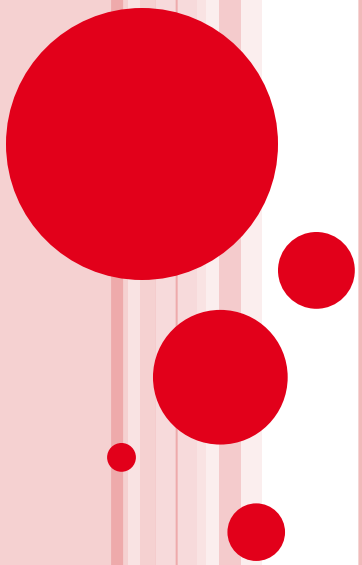
ITOUG
ITALIAN ORACLE USER GROUP

EVENTI | CONTENT | CHI SIAMO | COLLABORA CON NOI | ORGANIZZAZIONE | BLOG | DISCLAIMER

13:00 – 14:00	Pranzo | Lunch	
	(room "Alda Merini")	
14:00 – 14:45	**Paolo Zampolini** - Te… Trasformazione… architettura mu…	…lysis LLC | Oracle ACE Director …urt your database in new, unexpected, ways
15:00 – 15:45	**Niall Mc Phillips** What's Our Vecto… Oracle 23ai	…chnology Reply Automatizzare la documentazione di codice legacy con IA Generativa sfruttando i servizi OCI AI Vector Search
16:00 – 16:45	**Oren Nakdimon** - Moovit | Oracle ACE Director Write Less (Code) with More (Features)	**Piet de Visser** - PDVBV | Oracle ACE Pro Distributed databases, what to expect
17:00 – 17:45	**Neil Chandler** - Chandler Systems Ltd | Oracle ACE Director Fundamental Oracle Security – What Many Of You Are Not Doing	**Oren Nakdimon** - Moovit | Oracle ACE Director Oracle SQL Pattern Matching Made Easy
17:45 – 18:00	Conclusione | GoodBye Message	

Not for Eurovision Haters

ITOUG
ITALIAN ORACLE USER GROUP



THANK YOU☺

Oren Nakdimon

www.db-oriented.com

✉ oren@db-oriented.com

𝕏 @DBoriented