15-1: Creating Views

Vocabulary

1.view
2.view_name
3.force
4.simple view
5.noforce
6.create view
7.alias
8.subquery
9.complex view
9.replace

Try It / Solve It

1. Restrict access and display selective columns

Reduce complexity of queries from other internal systems. So, providing a way to view same data in a different manner.

Let the app code rely on views and allow the internal implementation of tables to be modified later.

2.
```
CREATE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

3.
```
SELECT*
FROM view_d_songs.
```

4.
```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist, d_songs.type_code
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

5.

```
CREATE OR REPLACE VIEW view_d_events_pkgs AS
SELECT evt.name "Name of Event", TO_CHAR(evt.event_date, 'dd-Month-yyyy')  "Event
date", thm.description "Theme description"
FROM  d_events  evt INNER JOIN d_themes  thm  ON evt.theme_code = thm.code
WHERE evt.event_date <= ADD_MONTHS(SYSDATE,1);
SELECT * FROM view_d_events_pkgs ;
```

6.

```
CREATE OR REPLACE VIEW view_min_max_avg_dpt_salary ("Department Id",
"Department Name", "Max Salary", "Min Salary", "Average Salary") AS
SELECT dpt.department_id, dpt.department_name, MAX(NVL(emp.salary,0)),
MIN(NVL(emp.salary,0)), ROUND(AVG(NVL(emp.salary,0)),2)
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id =
emp.department_id
GROUP BY (dpt.department_id, dpt.department_name);

select *
from view_min_max_avg_dpt_salary
```