EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?

data security and access control
simplifying complex queries
data consistency and logical independence

2. Create a simple view called view_d_songs that contains the ID, title and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

CREATE VIEW view-d-songs AS
SELECT id, title AS "Song Title", artist FROM d-songs
WHERE type-code = (SELECT id FROM d-types WHERE description = 'New Age');

3. SELECT * FROM view_d_songs. What was returned?

SELECT * FROM view-d-songs;

4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns.

Or use alias after the CREATE statement as shown.

CREATE OR REPLACE VIEW view-d-songs (songID, "Song Title", artist, Typecode) AS
SELECT id, title, artist, type-code FROM d-songs
WHERE type-code = (SELECT id FROM d-types
WHERE description = 'New Age');

5. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those planned for the coming months so he can make arrangements for each event's equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each column name.

```
CREATE OR REPLACE VIEW jason_events_vu AS
SELECT e.name AS "Event_Name", e.event_date AS
"Event_Date", t.description AS "Items" FROM d_events e
JOIN d_themes t ON e.theme_code = t.code;
```

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that displays the needed information for department managers.

```
CREATE OR REPLACE VIEW dept_salary_summary_
VU AS
SELECT department_id,
       MIN(salary) AS "min_salary",
       MAX(salary) AS "Max_Salary",
       AVG(salary) AS "AVG_salary"
FROM employees GROUP BY department_id;
```