2TPIF-E

Monticelli Leonardo

TASKS MODON 03.05

|  |  |
| --- | --- |
| **VIEW** | Identify the vocabulary word for each definition below. A subset of data from one or more tables that is generated from a query and stored as a virtual table. |
| **VIEW\_NAME** | Name of view |
| **FORCE** | Creates a view regardless of whether or not the base tables exist |
| **SIMPLE VIEW** | Derives data from a table, no functions or groups, performs DML operations through the view |
| **NOFORCE** | Creates the view only if the base table exists |
| **CREATE VIEW statement** | Statement used to create a new view |
| **Alias** | Specifies a name for each expression selected by the view’s query |
| **subquery** | A complete SELECT statement |
| **Complex View** | Derives data from more than one table, contains functions or groups of data, and does not always allow DML operations through the view |
| **REPLACE** | Re-creates the view if it already exists |

1. 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.

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';

2. SELECT \* FROM view\_d\_songs.

What was returned?

|  |  |  |
| --- | --- | --- |
| **FIRST\_NAME** | **LAST\_NAME** | **SALARY** |
| Steven | King | 24000 |
| Neena | Kochhar | 17000 |
| Lex | De Haan | 17000 |
| Jennifer | Whalen | 4400 |
| Shelley | Higgins | 12000 |
| William | Gietz | 8300 |
| Eleni | Zlotkey | 10500 |
| Ellen | Abel | 11000 |
| Jonathon | Taylor | 8600 |
| Kimberely | Grant | 7000 |

3. REPLACE view\_d\_songs. Add type\_code to the column list. Use aliases for all columns.

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';

4. Create a simple view that contains the first\_name, last\_name and the salary from Employees table.

CREATE view view\_d\_songs as SELECT first\_name, last\_name, salary FROM employees

5. Create a simple view that contains the first\_name, last\_name, and the salary whose surname begins with "K" (use Employees table).

CREATE view view\_d\_songs as SELECT first\_name, last\_name, salary FROM employees where last\_name like 'k%'

6. Create a simple view that contains the first\_name, last\_name and department\_id from Employees

table. Use aliases for each column name.

7. Create a view that displays the name of the event, the event date, and the theme description. Use

aliases for each column name.

8. Create a complex- view to display first\_name, last\_name from employees and department\_name

and location\_id from departments.Create a join between Employees and Departments using

department\_id.

9. Create a complex- view to display first\_name, last\_name and salary from employees and

department\_name and location from departments.Create a join between Employees and

Departments using department\_id and display the data for manager\_id=90.

10. Create a complex- view to display first\_name, last\_name and salary from employees and

department\_name and location from departments. Create a join between Employees and

Departments using department\_id and display the data for manager\_id=90 or manager\_id = 100.

11. 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.

12. Create a complex- view to display street\_address and the city from locations and department\_name and location\_id from departments.Create a join between Employees and Departments using location\_id.

13. Create a complex- view to display street\_address and the city from locations and department\_name and location\_id from departments.Create a join between Employees and Departments using location\_id and display the data for location\_id=1700.

14. Create a complex- view to display street\_address and the city from locations and department\_name and location\_id from departments.Create a join between Employees and Departments using location\_id and display the data for location\_id between 1700 and 2500.

15. The department managers need to know the total salaries, grouped by manager\_id. Use the Oracle database to prepare a view that displays the needed information for department managers. (use Employees table).