Lab 4: JOINS

Database Management System

Joins in SQL

Cross JOIN or Cartesian product

Syntax:

SELECT column-name-list
FROM
table-name1 CROSS JOIN table-name2;

INNER Join or EQUI Join:

Syntax:

SELECT column-name-list FROM

table-name1 INNER JOIN table-name2

ON table-name1.column-name = table-name2.column-name;

a) **Natural JOIN** (Not applicable in SQL server)

The syntax for Natural Join is,

*SELECT * FROM*

table-name1 NATURAL JOIN table-name2;

OUTER JOIN

i. Left Outer Join

Syntax:

SELECT column-name-list FROM

table-name1 LEFT OUTER JOIN table-name2

ON table-name1.column-name = table-name2.column-name;

ii. Right Outer Join:

Syntax:

SELECT column-name-list FROM

table-name1 RIGHT OUTER JOIN table-name2

ON table-name1.column-name = table-name2.column-name;

iii. Full Outer Join

Syntax:

SELECT column-name-list FROM

table-name1 FULL OUTER JOIN table-name2

ON table-name1.column-name = table-name2.column-name;

Q1. CREATE TWO TABLES tbl_info and tbl_details

tbl_info

ID	NAME
1	abhi
2	adam
4	alex

tbl_details

ID	ADDRESS
1	Ktm
2	Lalitpur
3	Bhaktapur

a. Apply CROSS JOIN on above table and display the result
SELECT * FROM tbl_info CROSS JOIN tbl_details;

b. Apply INNER JOIN on above table and display the result
SELECT * from tbl_info INNER JOIN tbl_details ON tbl_info.id = tbl_details.id;

c. Apply LEFT OUTER JOIN on above table and display the result
SELECT * FROM tbl_info LEFT OUTER JOIN tbl_details ON (tbl_info.id = tbl_details.id);

d. Apply RIGHT OUTER JOIN on above table and display the result
SELECT * FROM tbl_info RIGHT OUTER JOIN tbl_details ON (tbl_info.id = tbl_details.id);

e. Apply FULL OUTER JOIN on above table and display the result
SELECT * FROM tbl_info FULL OUTER JOIN tbl_details ON (tbl_info.id = tbl_details.id);

VIEWS

A view is a virtual table based on the result-set of an SQL statement.

Views are used for security purpose in databases, views restricts the user from viewing certain column and rows means by using view we can apply the restriction on accessing the particular rows and columns for specific user.

Write SQL for the following cases:

1. Create a database named '*views*'.
2. Use that database.
3. Create a table named *social_info* with column names *id, name, mobile_number, username and password.*
4. Insert the data as shown below:

id	name	mobile_number	username	password
1	Ronaldo	9801234568	ronaldo_007	Football66
2	Messi	9812234768	messi_10	Messi112
3	Pogba	9801245678	pogba_06	Pogba111
4	hazard	9711245679	hazard_7	hazard21
5	degea	9712245899	degea_01	degea11

5. Create a view named *social_media_view* & display name, username from it.

Syntax:

CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;

Example:

```
create view social_media_view as
select name, username
from social_media;

select * from social_media_view;
```

6. SQL Dropping a View

Syntax: DROP VIEW *view_name*;
Example: DROP VIEW social_media_view;