# **LAB ASSIGNMENT 9**

1. Create a view “engineers” containing details of employees working as engineers. The view, engineers should have emp\_id, f\_name, l\_name, salary and d\_name attributes.

🡺 CREATE VIEW engineers AS SELECT emp\_id, f\_name, l\_name, salary, d\_name FROM employee WHERE job\_type = 'Engineer';

1. Create a view “manager” containing emp\_id as ID, f\_name as name, and annual salary as ANNSAL for employees who work as managers.

🡺 CREATE VIEW manager AS SELECT emp\_id AS ID, f\_name AS name, salary \* 12 AS ANNSAL FROM employee WHERE job\_type = 'Manager';

1. Modify the view “manager” -- make the attribute ‘name’ a combination of f\_name and l\_name, and add d\_name attribute as ‘department’.

🡺 CREATE OR REPLACE VIEW manager AS SELECT emp\_id AS ID, f\_name || ' ' || l\_name AS name, d\_name AS department, salary \* 12 AS ANNSAL FROM employee WHERE job\_type = 'Manager';

1. Create view ‘dept\_wise’ with attributes name,minsal,maxsal, and avgsal, containing information of d\_name and department wise minimum salary, maximum salary and average salary.

🡺 CREATE VIEW dept\_wise AS SELECT d\_name AS name, MIN(salary) AS minsal, MAX(salary) AS maxsal, AVG(salary) AS avgsal FROM employee GROUP BY d\_name;

1. Create a view “emp\_location” with attribute name and location having information about the employees f\_name and their department’s location.

🡺 CREATE VIEW emp\_location AS SELECT e.f\_name AS name, d.d\_loc AS location FROM employee e INNER JOIN department d ON e.d\_name = d.d\_name;

1. Add attribute job\_type as ‘job’ in the emp\_location view where the tuples should be in ascending order of job\_type.

🡺 CREATE OR REPLACE VIEW emp\_location AS SELECT e.f\_name AS name, d.d\_loc AS location, e.job\_type AS job FROM employee e INNER JOIN department d ON e.d\_name = d.d\_name ORDER BY e.job\_type;

1. Create a view ‘emp5’ with f\_name as name and salary as ‘sal’ from the employee table.

🡺 CREATE VIEW emp5 AS SELECT f\_name AS name, salary AS sal FROM employee;

1. Show the schema description of the emp5 view.

🡺 SELECT \* FROM emp5;

1. Show the values from the emp5 view.

🡺 SELECT \* FROM emp5;

1. Find the salary of emma.

It can be done in two different ways:

select sal from emp5 where name=’emma’; Or select salary from employee where f\_name=’emma’;

🡺 SELECT sal FROM emp5 WHERE name = 'Emma';

1. Update emp5 view, increase the salary of emma, and make it 77000.

🡺 UPDATE emp5 SET sal = 77000 WHERE name = 'Emma';

1. Check if emma’s salary has been updated in the emp5 view.

🡺 SELECT \* FROM emp5 WHERE name = 'Emma';

1. Check if emma’s salary has been updated in the parent employee table.

🡺 SELECT salary FROM employee WHERE f\_name = 'Emma';

1. Update the employee table, increase the salary of emma, and make it 177000.

🡺 UPDATE employee SET salary = 177000 WHERE f\_name = 'Emma';

1. Check if the change is reflected in the emp5 view.

🡺 SELECT \* FROM emp5 WHERE name = 'Emma';

1. Update the view emp\_location. Change the job\_type of Saul from engineer to ‘COO’.

When can a view be updated?

1. The view is defined based on one and only one table.
2. The view should not have any field made out of aggregate functions.
3. The view must not have any DISTINCT clause in its definition.
4. The view must not have any GROUP BY or HAVING clause in its definition.
5. The view must not have any SUBQUERIES in its definitions.
6. If the view you want to update is based upon another view, the later should be updatable.

🡺 UPDATE emp\_location SET job = 'COO' WHERE name = 'Saul';

1. Create a table emp8 with attribute id and name.

🡺 CREATE TABLE emp8 (id INT, name VARCHAR(50));

1. Create a synonym (named - emp08) for emp8.

🡺 CREATE SYNONYM emp08 FOR emp8;

1. Describe emp8 and emp08.

🡺 DESCRIBE emp8;

DESCRIBE emp08;

1. Insert a tuple into emp8.

🡺 INSERT INTO emp8 VALUES (101, 'John');

1. Display all tuples from emp08.

🡺 SELECT \* FROM emp08;

1. Insert a tuple into emp08.

🡺 INSERT INTO emp08 VALUES (101, 'John');

1. Display all tuples from emp8.

🡺 SELECT \* FROM emp8;

1. Add a column dept to emp8 table.

🡺 ALTER TABLE emp8 ADD dept VARCHAR(50);

1. Describe emp08.

🡺 DESCRIBE emp08;

1. Delete all tuples from emp08.

🡺 DELETE FROM emp08;

1. Rename emp8 to employee8.

🡺 ALTER TABLE emp8 RENAME TO employee8;

1. Describe emp008.

🡺 DESCRIBE emp008;

1. Drop emp008.

🡺 DROP TABLE emp008;

1. Create an index on the attribute id of emp table.

🡺 CREATE INDEX id\_index ON emp (id);

1. Drop the index created in question no. 30.

🡺 DROP INDEX id\_index;