**Week 10 Day 1 Lab Coding Challenges**

# 

create database banks;

use banks;

1. create table account\_details with the following attributes and insert the data.

acc\_id int PK NOT NULL,

first\_name varchar(50) NOT NULL,

last\_name varchar(50) NOT NULL,

ssn char(10) NOT NULL,

acc\_holder\_id int NOT NULL,

balance decimal(20,4) DEFAULT '0.0000',

DROP TABLE IF EXISTS `account\_details`;

CREATE TABLE IF NOT EXISTS `account\_details` (

`acc\_id` int(10) NOT NULL,

`first\_name` varchar(50) NOT NULL,

`last\_name` varchar(50) NOT NULL,

`ssn` char(10) NOT NULL,

`acc\_holder\_id` int(10) NOT NULL,

`balance` decimal(20,4) DEFAULT '0.0000',

PRIMARY KEY (`acc\_id`));

INSERT INTO `account\_details` (`acc\_id`, `acc\_holder\_id`, `balance`, `first\_name`, `last\_name`, `ssn`) VALUES

(1, 100, 132.1020, 'Joseph', 'Cane', '098765432'),

(2, 300, 4435.2030, 'Kim', 'Karry', '087654321'),

(3, 120, 2345223.6600, 'Jim', 'Anderson', '123456780'),

(4, 90, 98763.2300, 'Jessie', 'Thomson', '765432109'),

(5, 110, 34221.1000, 'Palak', 'Patel', '654321890'),

(6, 80, 7634.8000, 'Max', 'Jerrard', '456789012'),

(7, 10, 876964.7000, 'Peter', 'Koshnov', '512345670'),

(8, 110, 299876.6000, 'Monica', 'Irodov', '120088551'),

(9, 100, 7659809.5300, 'Petro', 'Jenkins Jr', '123456789'),

(10, 200, 111.1200, 'Jeff', 'Joshua', '765432189' );

select \* from account\_details;

1. Create table **id\_passwords** with the following attributes.

user\_id varchar(20),

passwords varchar(20));

Create table id\_passwords (

user\_id varchar(20), passwords varchar(20)

);

insert into id\_passwords values

('deborah\_a', 'pass123'),

('pique\_xav', '123789pix'),

('jenny\_fawx', '##\*\*000'),

('alpha\_m','infinity');

1. **Write a transactional query that transfers 1000 dollars from Monica's account to Joseph's account**

update account\_details set balance=balance-1000 where first\_name='Monica';

update account\_details set balance=balance+1000 where first\_name='Joseph';

1. **Suppose while writing the above query you update i.e. transfer 1000 dollars to Peter's account instead of Joseph's account.**

**Write a query to discard all the changes and end the transaction**

#set sql\_safe\_updates=0;

set autocommit=0;

select \* from account\_details;

update account\_details set balance=balance-1000 where first\_name='Monica';

update account\_details set balance=balance+1000 where first\_name='Peter';

select \* from account\_details;

rollback;

select \* from account\_details;

set autocommit=1;

-- ----------------------------------------------------

# Datasets Used: employee\_details.csv and department\_details.csv

**# import table and data from .csv file**

-- ----------------------------------------------------

1. **Create a view "details" that contains the columns employee\_id, first\_name, last\_name and the salary from the table "employee\_details".**

create view details as select employee\_id, first\_name, last\_name, salary from employee\_details;

select \* from details;

1. **Update the view "details" such that it contains the records from the columns employee\_id, first\_name, last\_name, salary, hire\_date and job\_id where job\_id is ‘IT\_PROG’.**

-- 2 solutions

#create or replace view details as select employee\_id, first\_name, last\_name, salary, hire\_date, job\_id from employee\_details where job\_id= 'IT\_PROG';

alter view details as select employee\_id, first\_name, last\_name, salary, hire\_date, job\_id from employee\_details where job\_id= 'IT\_PROG';

select \* from details;

1. **Create a view "check\_salary" that contains the records from the columns employee\_id, first\_name, last\_name, job\_id and salary from the table "employee\_details" where the salary should be greater than 50000.**

create view check\_salary as select employee\_id, first\_name, last\_name, job\_id, salary from employee\_details where salary > 50000;

select \* from check\_salary;

1. **Create a view "location\_details" that contains the records from the columns department\_name, manager\_id and the location\_id from the table "department\_details" where the department\_name is ‘Shipping’.**

create view location\_details as select department\_name, manager\_id, location\_id from department\_details where department\_name='Shipping';

select \* from location\_details;

1. **Create a view "salary\_range" such that it contains the records from the columns employee\_id, first\_name, last\_name, job\_id and salary from the table "employee\_details" where the salary is in the range (30000 to 50000).**

create view salary\_range as

select employee\_id, first\_name, last\_name, job\_id, salary from employee\_details where salary between 30000 and 50000;

select \* from salary\_range;

1. **Create a view "pattern\_matching" such that it contains the records from the columns employee\_id, first\_name, job\_id and salary from the table name "employee\_details" where first\_name ends with "l".**

create view pattern\_matching as

select employee\_id, first\_name, job\_id, salary from employee\_details where first\_name like '%i';

select \* from pattern\_matching;

1. **Drop multiple existing views "pattern\_matching", "salary" and "location\_details".**

drop view pattern\_matching, salary\_range,location\_details;

1. **Create a view "employee\_department" that contains the common records from the tables "employee\_details" and "department\_table".**

create view employee\_department as

select emp.employee\_id, emp.first\_name, emp.salary, emp.job\_id, dept.department\_name from employee\_details

as emp inner join department\_details as dept

on emp.employee\_id=dept.employee\_id;

select \* from employee\_department;

-- ----------------------------------------------------

# Dataset Used: admission\_predict.csv

**# import table and data from .csv file**

-- ----------------------------------------------------

1. **A university focuses only on SOP and LOR score and considers these scores of the students who have a research paper. Create a view for that university.**

create view slr\_focus as

select 'Serial\_no',SOP,LOR,Research from admission\_predict where research =1;

select \* from slr\_focus;