1. Write a MySQL query to find the addresses (location\_id, street\_address, city, state\_province, country\_name) of all the departments.

**Hint : Use NATURAL JOIN.**

-- Create the 'locations' table

CREATE TABLE locations (

location\_id INT PRIMARY KEY,

street\_address VARCHAR(255),

postal\_code VARCHAR(50),

city VARCHAR(100),

state\_province VARCHAR(100),

country\_id CHAR(2)

);

-- Insert data into the 'locations' table

INSERT INTO locations (location\_id, street\_address, postal\_code, city, state\_province, country\_id) VALUES

(1000, '1297 Via Cola di Rie', '989', 'Roma', NULL, 'IT'),

(1100, '93091 Calle della Te', '10934', 'Venice', NULL, 'IT'),

(1200, '2017 Shinjuku-ku', '1689', 'Tokyo', 'Tokyo Prefecture', 'JP'),

(1300, '9450 Kamiya-cho', '6823', 'Hiroshima', NULL, 'JP'),

(1400, '2014 Jabberwocky Rd', '26192', 'Southlake', 'Texas', 'US'),

(1500, '2011 Interiors Blvd', '99236', 'South San Francisco', 'California', 'US'),

(1600, '2007 Zagora St', '50090', 'South Brunswick', 'New Jersey', 'US'),

(1700, '2004 Charade Rd', '98199', 'Seattle', 'Washington', 'US'),

(1800, '147 Spadina Ave', 'M5V 2L7', 'Toronto', 'Ontario', 'CA'),

(1900, '6092 Boxwood St', 'YSW 9T2', 'Whitehorse', 'Yukon', 'CA'),

(2000, '40-5-12 Laogianggen', '190518', 'Beijing', NULL, 'CN'),

(2100, '1298 Vileparle (E)', '490231', 'Mumbai', 'Maharashtra', 'IN'),

(2200, '12-98 Victoria Street', '2901', 'Sydney', 'New South Wales', 'AU'),

(2300, '198 Clementi North', '540198', 'Singapore', NULL, 'SG'),

(2400, '8204 Arthur St', NULL, 'London', NULL, 'UK'),

(2500, 'Magdalen Centre, The', 'OX9 9ZB', 'Oxford', 'Oxfordshire', 'UK'),

(2600, '9702 Chester Road', '9629850293', 'Stretford', 'Greater Manchester', 'UK'),

(2700, 'Schwanthalerstr. 703', '80925', 'Munich', 'Bavaria', 'DE'),

(2800, 'Rua Frei Caneca 1360', '01307-002', 'Sao Paulo', 'Sao Paulo', 'BR'),

(2900, '20 Rue des Corps-Saints', '1730', 'Geneva', 'Geneve', 'CH'),

(3000, 'Murtenstrasse 921', '3095', 'Bern', 'BE', 'CH'),

(3100, 'Pieter Breughelstraat 90', '3029SK', 'Utrecht', 'Utrecht', 'NL'),

(3200, 'Mariano Escobedo 999', '11932', 'Mexico City', 'Distrito Federal', 'MX');

-- Create the 'countries' table

CREATE TABLE countries (

country\_id CHAR(2) PRIMARY KEY,

country\_name VARCHAR(100),

region\_id INT

);

-- Insert data into the 'countries' table

INSERT INTO countries (country\_id, country\_name, region\_id) VALUES

('AR', 'Argentina', 2),

('AU', 'Australia', 3),

('BE', 'Belgium', 1),

('BR', 'Brazil', 2),

('CA', 'Canada', 2),

('CH', 'Switzerland', 1),

('CN', 'China', 3),

('DE', 'Germany', 1),

('DK', 'Denmark', 1),

('EG', 'Egypt', 4),

('FR', 'France', 1),

('HK', 'HongKong', 3),

('IL', 'Israel', 4),

('IN', 'India', 3),

('IT', 'Italy', 1),

('JP', 'Japan', 3),

('KW', 'Kuwait', 4),

('MX', 'Mexico', 2),

('NG', 'Nigeria', 4),

('NL', 'Netherlands', 1),

('SG', 'Singapore', 3),

('UK', 'United Kingdom', 1),

('US', 'United States', 2),

('ZM', 'Zambia', 4),

('ZW', 'Zimbabwe', 4);

-- This SQL query selects specific columns from the 'locations' table after performing a natural join with the 'countries' table.

SELECT

location\_id, -- Selecting the 'location\_id' column from the result set.

street\_address, -- Selecting the 'street\_address' column from the result set.

city, -- Selecting the 'city' column from the result set.

state\_province, -- Selecting the 'state\_province' column from the result set.

country\_name -- Selecting the 'country\_name' column from the result set.

FROM

locations -- Specifying the 'locations' table.

NATURAL JOIN

countries; -- Performing a natural join with the 'countries' table based on any common columns.

1. Write a MySQL query to find the name (first\_name, last name), department ID and name of all the employees.

CREATE TABLE employees (

EMPLOYEE\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

EMAIL VARCHAR(50),

PHONE\_NUMBER VARCHAR(20),

HIRE\_DATE DATE,

JOB\_ID VARCHAR(10),

SALARY DECIMAL(8, 2),

COMMISSION\_PCT DECIMAL(2, 2),

MANAGER\_ID INT,

DEPARTMENT\_ID INT

);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (100, 'Steven', 'King', 'SKING', '515.123.4567', '2003-06-17', 'AD\_PRES', 24000.00, NULL, NULL, 90), (101, 'Neena', 'Kochhar', 'NKOCHHAR', '515.123.4568', '2005-09-21', 'AD\_VP', 17000.00, NULL, 100, 90), (102, 'Lex', 'De Haan', 'LDEHAAN', '515.123.4569', '2001-01-13', 'AD\_VP', 17000.00, NULL, 100, 90), (103, 'Alexander', 'Hunold', 'AHUNOLD', '590.423.4567', '2006-01-03', 'IT\_PROG', 9000.00, NULL, 102, 60), (104, 'Bruce', 'Ernst', 'BERNST', '590.423.4568', '2007-05-21', 'IT\_PROG', 6000.00, NULL, 103, 60), (105, 'David', 'Austin', 'DAUSTIN', '590.423.4569', '2005-06-25', 'IT\_PROG', 4800.00, NULL, 103, 60), (106, 'Valli', 'Pataballa', 'VPATABAL', '590.423.4560', '2006-02-05', 'IT\_PROG', 4800.00, NULL, 103, 60), (107, 'Diana', 'Lorentz', 'DLORENTZ', '590.423.5567', '2007-02-07', 'IT\_PROG', 4200.00, NULL, 103, 60);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (108, 'Nancy', 'Greenberg', 'NGREENBE', '515.124.4569', '2002-08-17', 'FI\_MGR', 12008.00, NULL, 101, 100), (109, 'Daniel', 'Faviet', 'DFAVIET', '515.124.4169', '2002-08-16', 'FI\_ACCOUNT', 9000.00, NULL, 108, 100), (110, 'John', 'Chen', 'JCHEN', '515.124.4269', '2005-09-28', 'FI\_ACCOUNT', 8200.00, NULL, 108, 100), (111, 'Ismael', 'Sciarra', 'ISCIARRA', '515.124.4369', '2005-09-30', 'FI\_ACCOUNT', 7700.00, NULL, 108, 100), (112, 'Jose Manuel', 'Urman', 'JMURMAN', '515.124.4469', '2006-03-07', 'FI\_ACCOUNT', 7800.00, NULL, 108, 100), (113, 'Luis', 'Popp', 'LPOPP', '515.124.4567', '2007-12-07', 'FI\_ACCOUNT', 6900.00, NULL, 108, 100), (114, 'Den', 'Raphaely', 'DRAPHEAL', '515.127.4561', '2002-12-07', 'PU\_MAN', 11000.00, NULL, 100, 30), (115, 'Alexander', 'Khoo', 'AKHOO', '515.127.4562', '2003-05-18', 'PU\_CLERK', 3100.00, NULL, 114, 30);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (116, 'Shelli', 'Baida', 'SBAIDA', '515.127.4563', '2005-12-24', 'PU\_CLERK', 2900.00, NULL, 114, 30), (117, 'Sigal', 'Tobias', 'STOBIAS', '515.127.4564', '2005-07-24', 'PU\_CLERK', 2800.00, NULL, 114, 30), (118, 'Guy', 'Himuro', 'GHIMURO', '515.127.4565', '2006-11-15', 'PU\_CLERK', 2600.00, NULL, 114, 30), (119, 'Karen', 'Colmenares', 'KCOLMENA', '515.127.4566', '2007-08-10', 'PU\_CLERK', 2500.00, NULL, 114, 30), (120, 'Matthew', 'Weiss', 'MWEISS', '650.123.1234', '2004-07-18', 'ST\_MAN', 8000.00, NULL, 100, 50), (121, 'Adam', 'Fripp', 'AFRIPP', '650.123.2234', '2005-04-10', 'ST\_MAN', 8200.00, NULL, 100, 50), (122, 'Payam', 'Kaufling', 'PKAUFLIN', '650.123.3234', '2003-05-01', 'ST\_MAN', 7900.00, NULL, 100, 50), (123, 'Shanta', 'Vollman', 'SVOLLMAN', '650.123.4234', '2005-10-10', 'ST\_MAN', 6500.00, NULL, 100, 50);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (124, 'Kevin', 'Mourgos', 'KMOURGOS', '650.123.5234', '2007-11-16', 'ST\_MAN', 5800.00, NULL, 100, 50), (125, 'Julia', 'Nayer', 'JNAYER', '650.124.1214', '2005-07-16', 'ST\_CLERK', 3200.00, NULL, 120, 50), (126, 'Irene', 'Mikkilineni', 'IMIKKILI', '650.124.1224', '2006-09-28', 'ST\_CLERK', 2700.00, NULL, 120, 50), (127, 'James', 'Landry', 'JLANDRY', '650.124.1334', '2007-01-14', 'ST\_CLERK', 2400.00, NULL, 120, 50), (128, 'Steven', 'Markle', 'SMARKLE', '650.124.1434', '2008-03-08', 'ST\_CLERK', 2200.00, NULL, 120, 50), (129, 'Laura', 'Bissot', 'LBISSOT', '650.124.5234', '2005-08-20', 'ST\_CLERK', 3300.00, NULL, 121, 50), (130, 'Mozhe', 'Atkinson', 'MATKINSO', '650.124.6234', '2005-10-30', 'ST\_CLERK', 2800.00, NULL, 121, 50), (131, 'James', 'Marlow', 'JAMRLOW', '650.124.7234', '2005-02-16', 'ST\_CLERK', 2500.00, NULL, 121, 50);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (132, 'TJ', 'Olson', 'TJOLSON', '650.124.8234', '2007-04-10', 'ST\_CLERK', 2100.00, NULL, 121, 50), (133, 'Jason', 'Mallin', 'JMALLIN', '650.127.1934', '2004-06-14', 'SA\_MAN', 3300.00, 0.40, 100, 80), (134, 'Michael', 'Rogers', 'MROGERS', '650.127.1834', '2006-08-26', 'SA\_MAN', 2900.00, 0.30, 100, 80), (135, 'Ki', 'Gee', 'KGEE', '650.127.1734', '2007-12-12', 'SA\_MAN', 2400.00, 0.25, 100, 80), (136, 'Hazel', 'Philtanker', 'HPHILTAN', '650.127.1634', '2008-02-06', 'SA\_MAN', 2200.00, 0.20, 100, 80), (137, 'Renske', 'Ladwig', 'RLADWIG', '650.121.1234', '2003-07-14', 'SA\_REP', 3600.00, 0.15, 133, 80), (138, 'Stephen', 'Stiles', 'SSTILES', '650.121.2034', '2005-10-26', 'SA\_REP', 3200.00, 0.10, 133, 80), (139, 'John', 'Seo', 'JSEO', '650.121.2019', '2006-02-12', 'SA\_REP', 2700.00, 0.05, 133, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (140, 'Joshua', 'Patel', 'JPATEL', '650.121.1834', '2006-04-06', 'SA\_REP', 2500.00, 0.05, 133, 80), (141, 'Trenna', 'Rajs', 'TRAJS', '650.121.1750', '2003-10-17', 'SA\_REP', 2900.00, 0.05, 133, 80), (142, 'Curtis', 'Davies', 'CDAVIES', '650.121.1554', '2005-01-29', 'SA\_REP', 3100.00, 0.10, 134, 80), (143, 'Randall', 'Matos', 'RMATOS', '650.121.1434', '2006-03-15', 'SA\_REP', 2600.00, 0.05, 134, 80), (144, 'Peter', 'Vargas', 'PVARGAS', '650.121.1334', '2006-07-09', 'SA\_REP', 2500.00, 0.05, 134, 80), (145, 'John', 'Russell', 'JRUSSEL', '011.44.1344.429268', '2004-10-01', 'SA\_REP', 14000.00, 0.40, 100, 80), (146, 'Karen', 'Partners', 'KPARTNER', '011.44.1344.429278', '2005-01-05', 'SA\_REP', 13500.00, 0.30, 100, 80), (147, 'Alberto', 'Errazuriz', 'AERRAZUR', '011.44.1344.429288', '2005-03-10', 'SA\_REP', 12000.00, 0.30, 100, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (148, 'Gerald', 'Cambrault', 'GCAMBRAU', '011.44.1344.429318', '2007-10-15', 'SA\_REP', 11000.00, 0.30, 100, 80), (149, 'Eleni', 'Zlotkey', 'EZLOTKEY', '011.44.1344.429338', '2008-01-29', 'SA\_REP', 10500.00, 0.20, 100, 80), (150, 'Peter', 'Tucker', 'PTUCKER', '011.44.1344.129268', '2003-01-30', 'SA\_REP', 10000.00, 0.30, 100, 80), (151, 'David', 'Bernstein', 'DBERNSTE', '011.44.1344.129278', '2005-03-24', 'SA\_REP', 9500.00, 0.25, 100, 80), (152, 'Peter', 'Hall', 'PHALL', '011.44.1344.129288', '2005-08-20', 'SA\_REP', 9000.00, 0.25, 100, 80), (153, 'Christopher', 'Olsen', 'COLSEN', '011.44.1344.129298', '2006-03-30', 'SA\_REP', 8000.00, 0.20, 100, 80), (154, 'Nanette', 'Cambrault', 'NCAMBRAU', '011.44.1344.619268', '2006-12-09', 'SA\_REP', 7500.00, 0.20, 100, 80), (155, 'Oliver', 'Tuvault', 'OTUVAULT', '011.44.1344.619278', '2007-11-23', 'SA\_REP', 7000.00, 0.15, 100, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (156, 'Janette', 'King', 'JANKING', '011.44.1344.619288', '2004-01-30', 'SA\_REP', 10000.00, 0.35, 100, 80), (157, 'Patrick', 'Sully', 'PSULLY', '011.44.1344.829268', '2004-03-04', 'SA\_REP', 9500.00, 0.35, 100, 80), (158, 'Allan', 'McEwen', 'AMCEWEN', '011.44.1344.829268', '2004-08-01', 'SA\_REP', 9000.00, 0.35, 100, 80), (159, 'Lindsey', 'Smith', 'LSMITH', '011.44.1344.829278', '2005-03-10', 'SA\_REP', 8000.00, 0.30, 100, 80), (160, 'Louise', 'Doran', 'LDORAN', '011.44.1344.829288', '2005-12-15', 'SA\_REP', 7500.00, 0.30, 100, 80), (161, 'Sarath', 'Sewall', 'SSEWALL', '011.44.1344.929268', '2006-11-03', 'SA\_REP', 7000.00, 0.25, 100, 80), (162, 'Clara', 'Vishney', 'CVISHNEY', '011.44.1344.529268', '2007-11-11', 'SA\_REP', 6500.00, 0.25, 100, 80), (163, 'Danielle', 'Greene', 'DGREENE', '011.44.1344.629268', '2002-03-19', 'SA\_REP', 6000.00, 0.20, 100, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (164, 'Mattea', 'Marvins', 'MMARVINS', '011.44.1344.629278', '2003-05-07', 'SA\_REP', 3600.00, 0.15, 145, 80), (165, 'David', 'Lee', 'DLEE', '011.44.1344.629288', '2005-01-19', 'SA\_REP', 3400.00, 0.15, 145, 80), (166, 'Sundar', 'Ande', 'SANDE', '011.44.1344.629299', '2006-03-24', 'SA\_REP', 3000.00, 0.10, 145, 80), (167, 'Amit', 'Banda', 'ABANDA', '011.44.1344.529299', '2006-05-21', 'SA\_REP', 2900.00, 0.10, 145, 80), (168, 'Lisa', 'Ozer', 'LOZER', '011.44.1344.529302', '2007-03-11', 'SA\_REP', 11500.00, 0.25, 146, 80), (169, 'Harrison', 'Bloom', 'HBLOOM', '011.44.1344.529306', '2006-03-23', 'SA\_REP', 10000.00, 0.20, 146, 80), (170, 'Tayler', 'Fox', 'TFOX', '011.44.1344.529309', '2006-01-24', 'SA\_REP', 9600.00, 0.20, 146, 80), (171, 'William', 'Smith', 'WSMITH', '011.44.1344.519301', '2007-02-23', 'SA\_REP', 7400.00, 0.15, 146, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (172, 'Elizabeth', 'Bates', 'EBATES', '011.44.1344.519322', '2007-03-24', 'SA\_REP', 7300.00, 0.15, 146, 80), (173, 'Sundita', 'Kumar', 'SKUMAR', '011.44.1344.519333', '2008-04-21', 'SA\_REP', 6100.00, 0.10, 146, 80), (174, 'Ellen', 'Abel', 'EABEL', '011.44.1343.929268', '2004-05-11', 'SA\_REP', 11000.00, 0.30, 100, 80), (175, 'Alyssa', 'Hutton', 'AHUTTON', '011.44.1343.929278', '2006-03-19', 'SA\_REP', 8800.00, 0.25, 100, 80), (176, 'Jonathon', 'Taylor', 'JTAYLOR', '011.44.1343.929288', '2006-06-24', 'SA\_REP', 8600.00, 0.20, 100, 80), (177, 'Jack', 'Livingston', 'JLIVING', '011.44.1343.929299', '2006-04-23', 'SA\_REP', 8400.00, 0.20, 100, 80), (178, 'Kimberely', 'Grant', 'KGRANT', '011.44.1343.929302', '2007-05-24', 'SA\_REP', 7000.00, 0.15, 100, 80), (179, 'Charles', 'Johnson', 'CJOHNSON', '011.44.1343.929306', '2008-01-04', 'SA\_REP', 6200.00, 0.10, 100, 80);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (180, 'Winston', 'Taylor', 'WTAYLOR', '650.507.9876', '2006-01-24', 'SH\_CLERK', 3200.00, NULL, 120, 50), (181, 'Jean', 'Fleaur', 'JFLEAUR', '650.507.9877', '2006-02-23', 'SH\_CLERK', 3100.00, NULL, 120, 50), (182, 'Martha', 'Sullivan', 'MSULLIVA', '650.507.9878', '2007-06-21', 'SH\_CLERK', 2500.00, NULL, 120, 50), (183, 'Girard', 'Geoni', 'GGEONI', '650.507.9879', '2008-02-03', 'SH\_CLERK', 2800.00, NULL, 120, 50), (184, 'Nandita', 'Sarchand', 'NSARCHAN', '650.509.1876', '2004-01-27', 'SH\_CLERK', 4200.00, NULL, 121, 50), (185, 'Alexis', 'Bull', 'ABULL', '650.509.2876', '2005-02-20', 'SH\_CLERK', 4100.00, NULL, 121, 50), (186, 'Julia', 'Dellinger', 'JDELLING', '650.509.3876', '2006-06-24', 'SH\_CLERK', 3400.00, NULL, 121, 50), (187, 'Anthony', 'Cabrio', 'ACABRIO', '650.509.4876', '2007-02-07', 'SH\_CLERK', 3000.00, NULL, 121, 50);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (188, 'Kelly', 'Chung', 'KCHUNG', '650.505.1876', '2005-06-14', 'SH\_CLERK', 3800.00, NULL, 122, 50), (189, 'Jennifer', 'Dilly', 'JDILLY', '650.505.2876', '2005-08-13', 'SH\_CLERK', 3600.00, NULL, 122, 50), (190, 'Timothy', 'Gates', 'TGATES', '650.505.3876', '2006-07-11', 'SH\_CLERK', 2900.00, NULL, 122, 50), (191, 'Randall', 'Perkins', 'RPERKINS', '650.505.4876', '2007-12-19', 'SH\_CLERK', 2500.00, NULL, 122, 50), (192, 'Sarah', 'Bell', 'SBELL', '650.501.1876', '2004-02-04', 'SH\_CLERK', 4000.00, NULL, 123, 50), (193, 'Britney', 'Everett', 'BEVERETT', '650.501.2876', '2005-03-03', 'SH\_CLERK', 3900.00, NULL, 123, 50), (194, 'Samuel', 'McCain', 'SMCCAIN', '650.501.3876', '2006-07-01', 'SH\_CLERK', 3200.00, NULL, 123, 50), (195, 'Vance', 'Jones', 'VJONES', '650.501.4876', '2007-03-17', 'SH\_CLERK', 2800.00, NULL, 123, 50);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (196, 'Alana', 'Walsh', 'AWALSH', '650.507.9811', '2006-04-24', 'SH\_CLERK', 3100.00, NULL, 124, 50), (197, 'Kevin', 'Feeney', 'KFEENEY', '650.507.9822', '2006-05-23', 'SH\_CLERK', 3000.00, NULL, 124, 50), (198, 'Donald', 'OConnell', 'DOCONNEL', '650.507.9833', '2007-06-21', 'SH\_CLERK', 2600.00, NULL, 124, 50), (199, 'Douglas', 'Grant', 'DGRANT', '650.507.9844', '2008-01-13', 'SH\_CLERK', 2400.00, NULL, 124, 50), (200, 'Jennifer', 'Whalen', 'JWHALEN', '515.123.4444', '2003-09-17', 'AD\_ASST', 4400.00, NULL, 101, 10), (201, 'Michael', 'Hartstein', 'MHARTSTE', '515.123.5555', '2004-02-17', 'MK\_MAN', 13000.00, NULL, 100, 20), (202, 'Pat', 'Fay', 'PFAY', '603.123.6666', '2005-08-17', 'MK\_REP', 6000.00, NULL, 201, 20);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (203, 'Susan', 'Mavris', 'SMAVRIS', '515.123.7777', '2002-06-07', 'HR\_REP', 6500.00, NULL, 101, 40), (204, 'Hermann', 'Baer', 'HBAER', '515.123.8888', '2002-06-07', 'PR\_REP', 10000.00, NULL, 101, 70);

INSERT INTO employees (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (205, 'Shelley', 'Higgins', 'SHIGGINS', '515.123.8080', '2002-06-07', 'AC\_MGR', 12000.00, NULL, 101, 110), (206, 'William', 'Gietz', 'WGIETZ', '515.123.8181', '2002-06-07', 'AC\_ACCOUNT', 8300.00, NULL, 205, 110);

-- Create the table

CREATE TABLE Departments (

DEPARTMENT\_ID INT PRIMARY KEY,

DEPARTMENT\_NAME VARCHAR(50),

MANAGER\_ID INT,

LOCATION\_ID INT

);

-- Insert records into the table

INSERT INTO Departments (DEPARTMENT\_ID, DEPARTMENT\_NAME, MANAGER\_ID, LOCATION\_ID) VALUES

(10, 'Administration', 200, 1700),

(20, 'Marketing', 201, 1800),

(30, 'Purchasing', 114, 1700),

(40, 'Human Resources', 203, 2400),

(50, 'Shipping', 121, 1500),

(60, 'IT', 103, 1400),

(70, 'Public Relations', 204, 2700),

(80, 'Sales', 145, 2500),

(90, 'Executive', 100, 1700),

(100, 'Finance', 108, 1700),

(110, 'Accounting', 205, 1700),

(120, 'Treasury', 0, 1700),

(130, 'Corporate Tax', 0, 1700),

(140, 'Control And Credit', 0, 1700),

(150, 'Shareholder Services', 0, 1700),

(160, 'Benefits', 0, 1700),

(170, 'Manufacturing', 0, 1700),

(180, 'Construction', 0, 1700),

(190, 'Contracting', 0, 1700),

(200, 'Operations', 0, 1700),

(210, 'IT Support', 0, 1700),

(220, 'NOC', 0, 1700),

(230, 'IT Helpdesk', 0, 1700),

(240, 'Government Sales', 0, 1700),

(250, 'Retail Sales', 0, 1700),

(260, 'Recruiting', 0, 1700),

(270, 'Payroll', 0, 1700);

**Solution:**

**-- This SQL query selects specific columns from the 'employees' table after performing an inner join with the 'departments' table using the 'department\_id' column.**

**SELECT**

**first\_name, -- Selecting the 'first\_name' column from the result set.**

**last\_name, -- Selecting the 'last\_name' column from the result set.**

**department\_id, -- Selecting the 'department\_id' column from the result set.**

**department\_name -- Selecting the 'department\_name' column from the result set.**

**FROM**

**employees -- Specifying the 'employees' table.**

**JOIN**

**departments -- Specifying the 'departments' table.**

**USING**

**(department\_id); -- Performing an inner join using the 'department\_id' column, which is common in both tables.**

1. Write a MySQL query to find the name (first\_name, last\_name), job, department ID and name of the employees who works in London.

**Sample table: employees**

**Sample table: departments**

**Sample table: locations**

-- This SQL query selects specific columns from the 'employees' and 'departments' tables, as well as the 'locations' table, to retrieve information about employees in the London city.

SELECT

e.first\_name, -- Selecting the 'first\_name' column from the 'employees' table and aliasing it as 'e'.

e.last\_name, -- Selecting the 'last\_name' column from the 'employees' table and aliasing it as 'e'.

e.job\_id, -- Selecting the 'job\_id' column from the 'employees' table and aliasing it as 'e'.

e.department\_id, -- Selecting the 'department\_id' column from the 'employees' table and aliasing it as 'e'.

d.department\_name -- Selecting the 'department\_name' column from the 'departments' table and aliasing it as 'd'.

FROM

employees e -- Specifying the 'employees' table and aliasing it as 'e'.

JOIN

departments d -- Specifying the 'departments' table and aliasing it as 'd'.

ON

(e.department\_id = d.department\_id) -- Performing a join between 'employees' and 'departments' based on the 'department\_id' column.

JOIN

locations l ON -- Joining the 'locations' table and aliasing it as 'l'.

(d.location\_id = l.location\_id) -- Performing a join between 'departments' and 'locations' based on the 'location\_id' column.

WHERE

LOWER(l.city) = 'London'; -- Filtering the result to only include rows where the city in lowercase is 'London'.

1. Write a MySQL query to find the employee id, name (last\_name) along with their manager\_id and name (last\_name).

**Sample table: employees**

-- This SQL query selects specific columns from the 'employees' table, twice aliased as 'e' and 'm', to retrieve information about employees and their managers.

SELECT

e.employee\_id 'Emp\_Id', -- Selecting the 'employee\_id' column from the 'employees' table, aliased as 'e', and renaming it as 'Emp\_Id'.

e.last\_name 'Employee', -- Selecting the 'last\_name' column from the 'employees' table, aliased as 'e', and renaming it as 'Employee'.

m.employee\_id 'Mgr\_Id', -- Selecting the 'employee\_id' column from the 'employees' table, aliased as 'm', and renaming it as 'Mgr\_Id'.

m.last\_name 'Manager' -- Selecting the 'last\_name' column from the 'employees' table, aliased as 'm', and renaming it as 'Manager'.

FROM

employees e -- Specifying the 'employees' table and aliasing it as 'e'.

JOIN

employees m -- Joining the 'employees' table again and aliasing it as 'm'.

ON

(e.manager\_id = m.employee\_id); -- Performing a join between 'employees' and itself based on the 'manager\_id' column to associate employees with their managers.

1. Write a MySQL query to find the name (first\_name, last\_name) and hire date of the employees who was hired after 'Jones'.

**Sample table: employees**

-- This SQL query selects specific columns from the 'employees' table to retrieve information about employees hired after an employee with the last name 'Jones'.

SELECT

e.first\_name, -- Selecting the 'first\_name' column from the 'employees' table.

e.last\_name, -- Selecting the 'last\_name' column from the 'employees' table.

e.hire\_date -- Selecting the 'hire\_date' column from the 'employees' table.

FROM

employees e -- Specifying the 'employees' table and aliasing it as 'e'.

JOIN

employees davies -- Joining the 'employees' table again and aliasing it as 'davies'.

ON

(davies.last\_name = 'Jones') -- Performing a join based on the condition where the last name in 'davies' is 'Jones'.

WHERE

davies.hire\_date < e.hire\_date; -- Filtering the result to include only employees hired after the employee with the last name 'Jones'.

1. Write a MySQL query to get the department name and number of employees in the department.

**Sample table: employees**

**Sample table: departments**

-- This SQL query retrieves the count of employees in each department, along with the department names, from the 'departments' and 'employees' tables.

SELECT

department\_name AS 'Department Name', -- Selecting the 'department\_name' column from the 'departments' table and aliasing it as 'Department Name'.

COUNT(\*) AS 'No of Employees' -- Counting the number of records (employees) in each department and aliasing it as 'No of Employees'.

FROM

departments -- Specifying the 'departments' table.

INNER JOIN

employees -- Performing an inner join with the 'employees' table.

ON

employees.department\_id = departments.department\_id -- Joining the 'employees' and 'departments' tables based on the 'department\_id' column.

GROUP BY

departments.department\_id, department\_name -- Grouping the result set by department ID and department name.

ORDER BY

department\_name; -- Ordering the result set by department name in ascending order.

**7.** Write a MySQL query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.

**Sample table: employees**

-- This SQL query retrieves specific columns from the 'job\_history' table and calculates the duration of each job in days for employees in the specified department.

SELECT

employee\_id, -- Selecting the 'employee\_id' column from the result set.

job\_title, -- Selecting the 'job\_title' column from the result set.

end\_date - start\_date AS Days -- Calculating the difference between 'end\_date' and 'start\_date' columns and aliasing it as 'Days'.

FROM

job\_history -- Specifying the 'job\_history' table.

NATURAL JOIN

jobs -- Performing a natural join with the 'jobs' table.

WHERE

department\_id = 90; -- Filtering the result to include only records where the department ID is 90.

**8.** Write a MySQL query to display the department ID and name and first name of manager.

**Sample table: employees**

**Sample table: departments**

-- This SQL query retrieves specific columns from the 'departments' and 'employees' tables to get information about department managers.

SELECT

d.department\_id, -- Selecting the 'department\_id' column from the 'departments' table.

d.department\_name, -- Selecting the 'department\_name' column from the 'departments' table.

d.manager\_id, -- Selecting the 'manager\_id' column from the 'departments' table.

e.first\_name -- Selecting the 'first\_name' column from the 'employees' table and aliasing it as 'e'.

FROM

departments d -- Specifying the 'departments' table and aliasing it as 'd'.

INNER JOIN

employees e -- Performing an inner join with the 'employees' table and aliasing it as 'e'.

ON

(d.manager\_id = e.employee\_id); -- Joining the 'departments' and 'employees' tables based on the 'manager\_id' column to associate departments with their managers.

**9.** Write a MySQL query to display the department name, manager name, and city.

**Sample table: employees**

**Sample table: departments**

**Sample table: locations**

-- This SQL query retrieves specific columns from the 'departments', 'employees', and 'locations' tables to get information about department managers and their corresponding locations.

SELECT

d.department\_name, -- Selecting the 'department\_name' column from the 'departments' table.

e.first\_name, -- Selecting the 'first\_name' column from the 'employees' table and aliasing it as 'e'.

l.city -- Selecting the 'city' column from the 'locations' table and aliasing it as 'l'.

FROM

departments d -- Specifying the 'departments' table and aliasing it as 'd'.

JOIN

employees e -- Performing a join with the 'employees' table and aliasing it as 'e'.

ON

(d.manager\_id = e.employee\_id) -- Joining the 'departments' and 'employees' tables based on the 'manager\_id' column to associate departments with their managers.

JOIN

locations l USING (location\_id); -- Performing a join with the 'locations' table based on the 'location\_id' column.

**10.** Write a MySQL query to display the job title and average salary of employees.

**Sample table: employees**

**Sample table: jobs**

-- Create the table

CREATE TABLE Jobs (

JOB\_ID VARCHAR(10) PRIMARY KEY,

JOB\_TITLE VARCHAR(50),

MIN\_SALARY INT,

MAX\_SALARY INT

);

-- Insert records into the table

INSERT INTO Jobs (JOB\_ID, JOB\_TITLE, MIN\_SALARY, MAX\_SALARY) VALUES

('AD\_PRES', 'President', 20000, 40000),

('AD\_VP', 'Administration Vice President', 15000, 30000),

('AD\_ASST', 'Administration Assistant', 3000, 6000),

('FI\_MGR', 'Finance Manager', 8200, 16000),

('FI\_ACCOUNT', 'Accountant', 4200, 9000),

('AC\_MGR', 'Accounting Manager', 8200, 16000),

('AC\_ACCOUNT', 'Public Accountant', 4200, 9000),

('SA\_MAN', 'Sales Manager', 10000, 20000),

('SA\_REP', 'Sales Representative', 6000, 12000),

('PU\_MAN', 'Purchasing Manager', 8000, 15000),

('PU\_CLERK', 'Purchasing Clerk', 2500, 5500),

('ST\_MAN', 'Stock Manager', 5500, 8500),

('ST\_CLERK', 'Stock Clerk', 2000, 5000),

('SH\_CLERK', 'Shipping Clerk', 2500, 5500),

('IT\_PROG', 'Programmer', 4000, 10000),

('MK\_MAN', 'Marketing Manager', 9000, 15000),

('MK\_REP', 'Marketing Representative', 4000, 9000),

('HR\_REP', 'Human Resources Representative', 4000, 9000),

('PR\_REP', 'Public Relations Representative', 4500, 10500);

Sol: -- This SQL query calculates the average salary for each job title by joining the 'employees' and 'jobs' tables.

SELECT

job\_title, -- Selecting the 'job\_title' column from the result set.

AVG(salary) -- Calculating the average salary and selecting it from the result set.

FROM

employees -- Specifying the 'employees' table.

NATURAL JOIN

jobs -- Performing a natural join with the 'jobs' table.

GROUP BY

job\_title; -- Grouping the result set by job title to calculate the average salary for each job.

**11.** Write a MySQL query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job.

**Sample table: employees**

**Sample table: jobs**

-- This SQL query selects specific columns from the 'employees' and 'jobs' tables and calculates the difference between each employee's salary and the minimum salary for their job title.

SELECT

job\_title, -- Selecting the 'job\_title' column from the result set.

first\_name, -- Selecting the 'first\_name' column from the result set.

salary - min\_salary AS 'Salary - Min\_Salary' -- Calculating the difference between the salary and the minimum salary for each job title and aliasing it as 'Salary - Min\_Salary'.

FROM

employees -- Specifying the 'employees' table.

NATURAL JOIN

jobs; -- Performing a natural join with the 'jobs' table.

**12.** Write a MySQL query to display the job history that were done by any employee who is currently drawing more than 10000 of salary.

**Sample table: employees**

**Sample table: Job\_history**

-- Create the table

CREATE TABLE Job\_history (

EMPLOYEE\_ID INT,

START\_DATE DATE,

END\_DATE DATE,

JOB\_ID VARCHAR(10),

DEPARTMENT\_ID INT

);

-- Insert records into the table

INSERT INTO Job\_history (EMPLOYEE\_ID, START\_DATE, END\_DATE, JOB\_ID, DEPARTMENT\_ID) VALUES

(102, '1993-01-13', '1998-07-24', 'IT\_PROG', 60),

(101, '1989-09-21', '1993-10-27', 'AC\_ACCOUNT', 110),

(101, '1993-10-28', '1997-03-15', 'AC\_MGR', 110),

(201, '1996-02-17', '1999-12-19', 'MK\_REP', 20),

(114, '1998-03-24', '1999-12-31', 'ST\_CLERK', 50),

(122, '1999-01-01', '1999-12-31', 'ST\_CLERK', 50),

(200, '1987-09-17', '1993-06-17', 'AD\_ASST', 90),

(176, '1998-03-24', '1998-12-31', 'SA\_REP', 80),

(176, '1999-01-01', '1999-12-31', 'SA\_MAN', 80),

(200, '1994-07-01', '1998-12-31', 'AC\_ACCOUNT', 90);

-- This SQL query retrieves all columns from the 'job\_history' table for employees whose salary is greater than 10000.

SELECT

jh.\* -- Selecting all columns from the 'job\_history' table.

FROM

job\_history jh -- Specifying the 'job\_history' table and aliasing it as 'jh'.

JOIN

employees e -- Performing a join with the 'employees' table and aliasing it as 'e'.

ON

(jh.employee\_id = e.employee\_id) -- Joining the 'job\_history' and 'employees' tables based on the 'employee\_id' column to associate job history with employees.

WHERE

salary > 10000; -- Filtering the result to include only records where the salary is greater than 10000.

**13.** Write a MySQL query to display department name, name (first\_name, last\_name), hire date, salary of the manager for all managers whose experience is more than 15 years.

**Sample table: employees**

**Sample table: departments**

-- This SQL query selects specific columns from the 'departments' and 'employees' tables to retrieve information about department managers with more than 15 years of experience.

SELECT

first\_name, -- Selecting the 'first\_name' column from the result set.

last\_name, -- Selecting the 'last\_name' column from the result set.

hire\_date, -- Selecting the 'hire\_date' column from the result set.

salary, -- Selecting the 'salary' column from the result set.

(DATEDIFF(now(), hire\_date))/365 Experience -- Calculating the experience in years and aliasing it as 'Experience'.

FROM

departments d -- Specifying the 'departments' table and aliasing it as 'd'.

JOIN

employees e -- Joining the 'employees' table and aliasing it as 'e'.

ON

(d.manager\_id = e.employee\_id) -- Joining the 'departments' and 'employees' tables based on the 'manager\_id' column to associate department managers with their departments.

WHERE

(DATEDIFF(now(), hire\_date))/365 > 15; -- Filtering the result to include only records where the experience in years is greater than 15.