CREATE DATABASE w3q1\_orders;

USE w3q1\_orders;

CREATE TABLE Salesman(

salesman\_id int,

sname varchar(50),

city varchar(50),

commission float

);

INSERT INTO

Salesman(salesman\_id, sname, city, commission)

VALUES

(5001, 'James Hoog', 'New York', 0.15),

(5002, 'Nail Knite', 'Paris', 0.13),

(5005, 'Pit Alex', 'London', 0.11),

(5006, 'Mc Lyon', 'Paris', 0.14),

(5007, 'Paul Adam', 'Rome', 0.13),

(5003, 'Lauson Hen', 'San Jose', 0.12);

CREATE TABLE Customer(

customer\_id int,

cust\_name varchar(50),

city varchar(25),

grade int,

salesman\_id int

);

INSERT INTO

Customer(customer\_id, cust\_name, city, grade, salesman\_id)

VALUES

(3002, 'Nick Rimando', 'New York', 100, 5001),

(3007, 'Brad Davis', 'New York', 200, 5001),

(3005, 'Graham Zusi', 'California', 200, 5002),

(3008, 'Julian Green', 'London', 300, 5002),

(3004, 'Fabian Johnson', 'Paris', 300, 5006),

(3009, 'Geoff Cameron', 'Berlin', 100, 5003),

(3003, 'Jozy Altidor', 'Moscow', 200, 5007),

(3001, 'Brad Guzan', 'London', 300, 5005);

CREATE TABLE Orders(

ord\_no int,

purch\_amt double,

ord\_date date,

salesman\_id int,

customer\_id int

);

INSERT INTO

Orders(

ord\_no,

purch\_amt,

ord\_date,

customer\_id,

salesman\_id

)

VALUES

(70001, 150.5 , '2012-10-05', 3005, 5002),

(70009, 270.65, '2012-09-10', 3001, 5005),

(70002, 65.26, '2012-10-05', 3002, 5001),

(70004, 110.5 , '2012-08-17', 3009, 5003),

(70007, 948.5 , '2012-09-10', 3005, 5002),

(70005, 2400.6 , '2012-07-27', 3007, 5001),

(70008, 5760 , '2012-09-10', 3002, 5001),

(70010, 1983.43, '2012-10-10', 3004, 5006),

(70003, 2480.4 , '2012-10-10', 3009, 5003),

(70012, 250.45, '2012-06-27', 3008, 5002),

(70011, 75.29, '2012-08-17', 3003, 5007),

(70013, 3045.6 , '2012-04-25', 3002, 5001);

/\*SELECT

\*

FROM

Orders; \*/

-- 1a , commission > 12%

SELECT

Customer.cust\_name AS "Customer Name",

Customer.city,

Salesman.sname AS "Salesman",

Salesman.commission

FROM

Customer

INNER JOIN Salesman ON Customer.salesman\_id = Salesman.salesman\_id

WHERE

Salesman.commission > .12;

-- 1b

SELECT

Customer.cust\_name,

Customer.city,

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt AS "Order Amount"

FROM

Customer

LEFT OUTER JOIN Orders ON Customer.customer\_id = Orders.customer\_id

order by

Orders.ord\_date;

-- 1c

SELECT

Customer.cust\_name,

Customer.city,

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt AS "Order Amount",

Salesman.sname,

Salesman.commission

FROM

Customer

LEFT OUTER JOIN Orders ON Customer.customer\_id = Orders.customer\_id

LEFT OUTER JOIN Salesman ON Salesman.salesman\_id = Orders.salesman\_id;

-- 1d

SELECT

Customer.cust\_name,

Customer.city,

Customer.grade,

Salesman.sname AS "Salesman",

Salesman.city

FROM

Customer

RIGHT OUTER JOIN Salesman ON Salesman.salesman\_id = Customer.salesman\_id

ORDER BY

Salesman.salesman\_id;

-- 1e

SELECT

\*

FROM

Salesman

CROSS JOIN Customer

-- WHERE Customer.salesman\_id = Salesman.salesman\_id

;

-- 1f

SELECT

\*

FROM

Salesman

CROSS JOIN Customer

WHERE

Salesman.city = Customer.city;

-- 1g

SELECT

Customer.cust\_name,

Customer.city,

Customer.grade,

Salesman.sname AS "Salesman",

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt

FROM

Customer

RIGHT OUTER JOIN Salesman ON Salesman.salesman\_id = Customer.salesman\_id

LEFT OUTER JOIN Orders ON Orders.customer\_id = Customer.customer\_id

WHERE

Orders.purch\_amt >= 2000

AND Customer.grade IS NOT NULL;

DELETE FROM Orders;

DELETE FROM Customer;

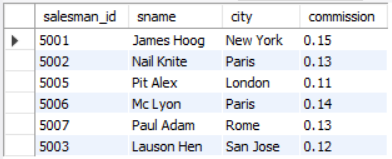
DELETE FROM Salesman;

DROP TABLE Salesman, Customer, Orders;

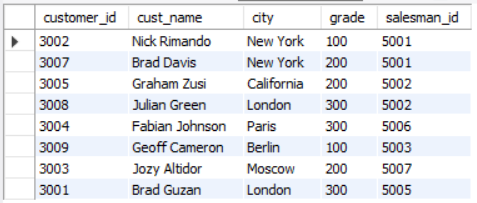
DROP DATABASE w3q1\_orders;

Inputs:

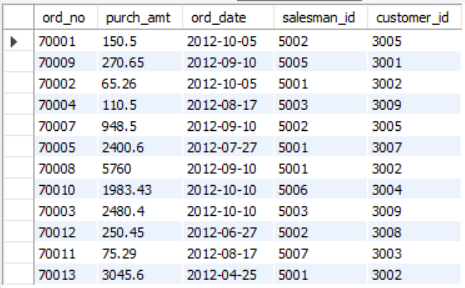
* Salesman table



* Customer table



* Orders



Queries & Outputs:

* SELECT

Customer.cust\_name AS "Customer Name",

Customer.city,

Salesman.sname AS "Salesman",

Salesman.commission

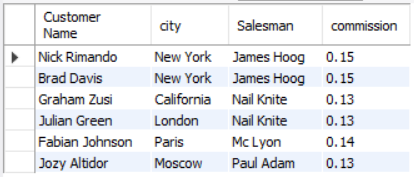
FROM

Customer

INNER JOIN Salesman ON Customer.salesman\_id = Salesman.salesman\_id

WHERE

Salesman.commission > .12;



* SELECT

Customer.cust\_name,

Customer.city,

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt AS "Order Amount"

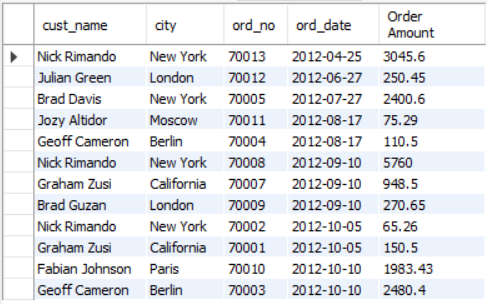
FROM

Customer

LEFT OUTER JOIN Orders ON Customer.customer\_id = Orders.customer\_id

order by

Orders.ord\_date;



* SELECT

Customer.cust\_name,

Customer.city,

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt AS "Order Amount",

Salesman.sname,

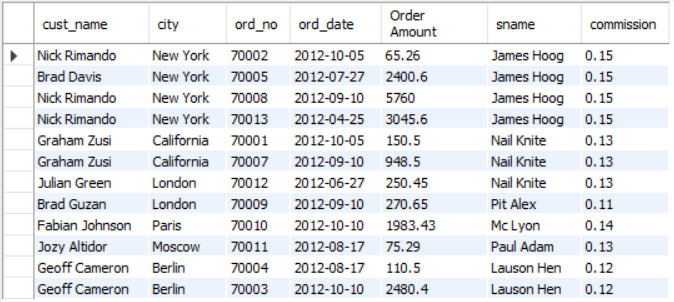
Salesman.commission

FROM

Customer

LEFT OUTER JOIN Orders ON Customer.customer\_id = Orders.customer\_id

LEFT OUTER JOIN Salesman ON Salesman.salesman\_id = Orders.salesman\_id;



* SELECT

Customer.cust\_name,

Customer.city,

Customer.grade,

Salesman.sname AS "Salesman",

Salesman.city

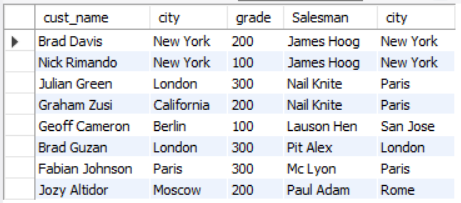
FROM

Customer

RIGHT OUTER JOIN Salesman ON Salesman.salesman\_id = Customer.salesman\_id

ORDER BY

Salesman.salesman\_id;

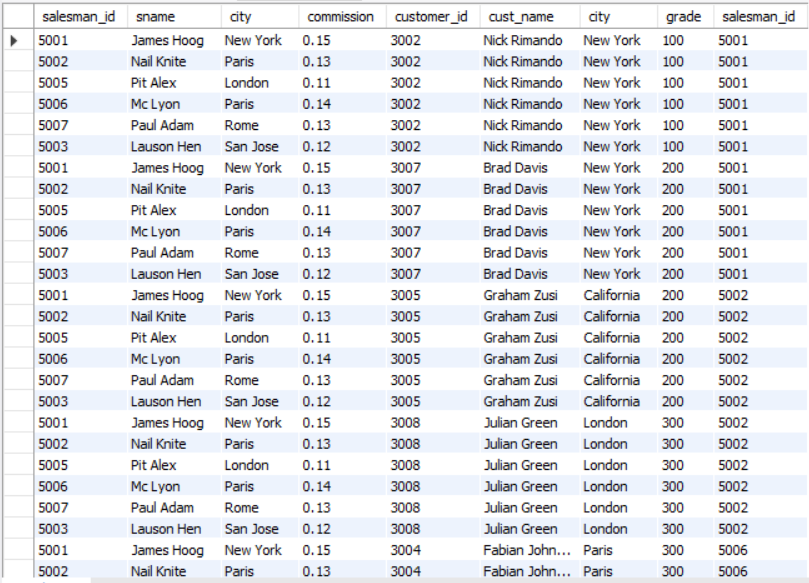


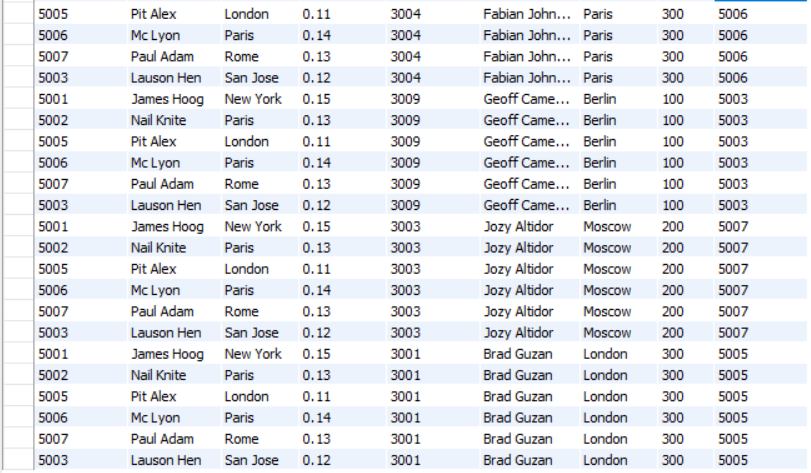
* SELECT \*

FROMSalesman

CROSS JOIN Customer

-- WHERE Customer.salesman\_id = Salesman.salesman\_id;





-> SELECT

\*

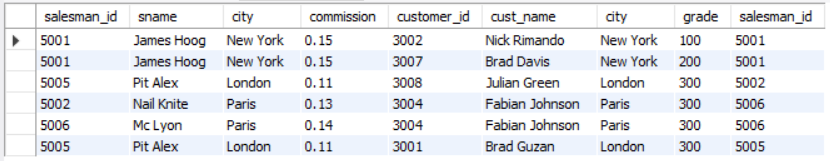
FROM

Salesman

CROSS JOIN Customer

WHERE

Salesman.city = Customer.city;



* SELECT

Customer.cust\_name,

Customer.city,

Customer.grade,

Salesman.sname AS "Salesman",

Orders.ord\_no,

Orders.ord\_date,

Orders.purch\_amt

FROM

Customer

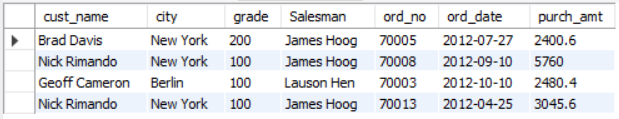
RIGHT OUTER JOIN Salesman ON Salesman.salesman\_id = Customer.salesman\_id

LEFT OUTER JOIN Orders ON Orders.customer\_id = Customer.customer\_id

WHERE

Orders.purch\_amt >= 2000

AND Customer.grade IS NOT NULL;



2)

Code:

CREATE DATABASE Empire;

USE Empire;

CREATE TABLE Employee (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(25) NOT NULL,

last\_name VARCHAR(25) NOT NULL,

email VARCHAR(25) NOT NULL,

phone\_number VARCHAR(12) NOT NULL,

hire\_date DATE NOT NULL,

job\_id VARCHAR(13) NOT NULL,

salary FLOAT(7,2) NOT NULL DEFAULT 0.0,

commission\_pct FLOAT(7,2) /\*NOT NULL\*/ DEFAULT 0,

manager\_id INT NOT NULL,

department\_id INT NOT NULL -- ,

-- CONSTRAINT manager\_id CHECK manager\_id IN employee\_id

);

INSERT INTO Employee

VALUES

(100, "Steven", "King", "SKING", '515.123.4567', '2003-06-17', 'AD\_PRES', 24000.00, 0.00, 0, 90),

(101, "Neena", "Kochhar", "NKOCCHAR", '515.123.4568', '2005-09-21', 'AD\_VP', 17000.00, 0.00, 100, 40),

(102, "Lex", "De Haan", "LDEHAAN", '515.123.4569', '2001-01-13', 'AD\_VP', 17000.00, 0.00, 100, 90),

(103, "Alexander", "Hunold", "AHUNOLD", '590.423.4567', '2006-01-03', 'IT\_PROG', 9000.00, 0.00, 102, 60),

(104, "Bruce", "Ernst", "BERNST", '590.423.4568', '2007-05-21', 'IT\_PROG', 6000.00, 0.00, 103, 60),

(105, "David", "Austin", "DAUSTIN", '590.423.4569', '2005-06-25', 'IT\_PROG', 4800.00, 0.00, 103, 40),

(106, "Valli", "Pataballa", "VPATABAL", '590.423.4560', '2006-02-05', 'IT\_PROG', 4800.00, 0.00, 103, 60),

(107, "Diana", "Lorentz", "DLORENTZ", '590.423.5567', '2007-02-07', 'IT\_PROG', 4800.00, 0.00, 103, 60),

(108, "Nancy", "Greenberg", "NGREENBE", '515.124.4569', '2002-08-17', 'FI\_MGR', 12008.00, 0.00, 103, 100),

(109, "Daniel", "Faviet", "DFAVIET", '515.124.4169', '2002-08-16', 'FI\_ACCOUNT', 9000.00, 0.00, 103, 100),

(110, "John", "Chen", "JCHEN", '515.124.4269', '2005-09-28', 'FI\_ACCOUNT', 8200.00, 0.00, 103, 40)

;

-- 2a

SELECT first\_name, last\_name, department\_id

FROM Employee

WHERE employee\_id IN (

SELECT employee\_id

FROM Employee

WHERE SALARY > 3.7E3

);

-- 2a 2nd way

/\* SELECT first\_name, last\_name, department\_id

FROM Employee

WHERE salary > 3700

; /\* why only 3700, not 5k? \*/

-- 2b

SELECT department\_id, SUM(salary) AS Total\_Salary

FROM Employee

WHERE department\_id IN (

SELECT DISTINCT department\_id

FROM Employee

-- GROUP BY department\_id

-- HAVING COUNT(department\_id) >= 1

)

GROUP BY department\_id

HAVING COUNT(department\_id) >= 1

;

-- 2b 2nd way

/\*SELECT department\_id, SUM(salary) AS Total\_Salary

FROM Employee

GROUP BY department\_id

-- HAVING COUNT(department\_id) >= 1

; /\*\*/

-- SELECT DISTINCT department\_id FROM Employee;

-- 2c but no nested query? , Insert meme chimp asks "where nested query?"

SELECT DISTINCT department\_id

FROM Employee

WHERE department\_id IN (

SELECT department\_id

FROM Employee

)

GROUP BY department\_id

HAVING COUNT(department\_id) >= 1

;

-- 2d

SELECT first\_name, last\_name, department\_id

FROM Employee

WHERE salary > (

SELECT AVG(salary)

FROM Employee

);

-- 2e

SELECT first\_name, last\_name, salary, department\_id

FROM Employee

WHERE salary > (

SELECT MAX(salary)

FROM Employee

WHERE department\_id = 40

);

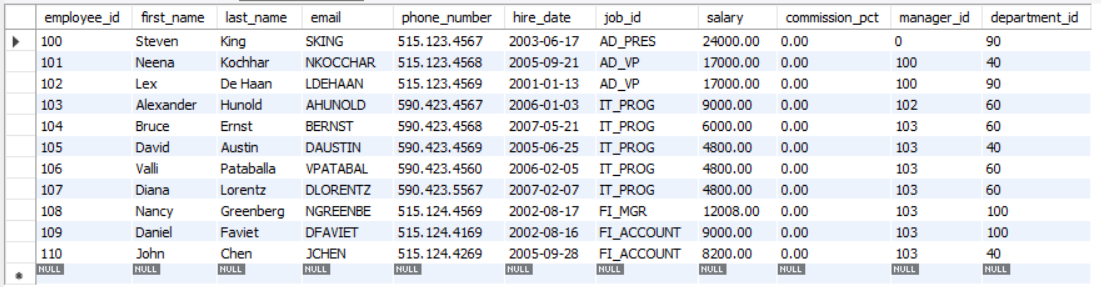
DELETE FROM Employee;

DROP TABLE Employee;

DROP DATABASE Empire;

Inputs:

* Employee Table



Queries & Outputs:

* SELECT first\_name, last\_name, department\_id

FROM Employee

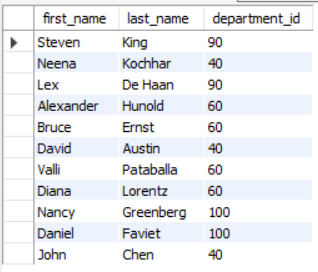
WHERE employee\_id IN (

SELECT employee\_id

FROM Employee

WHERE SALARY > 3.7E3

);



* SELECT department\_id, SUM(salary) AS Total\_Salary

FROM Employee

WHERE department\_id IN (

SELECT DISTINCT department\_id

FROM Employee

-- GROUP BY department\_id

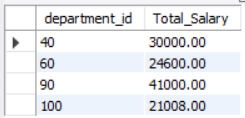
-- HAVING COUNT(department\_id) >= 1

)

GROUP BY department\_id

HAVING COUNT(department\_id) >= 1

;



* SELECT DISTINCT department\_id

FROM Employee

WHERE department\_id IN (

SELECT department\_id

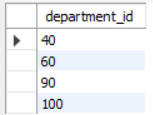
FROM Employee

)

GROUP BY department\_id

HAVING COUNT(department\_id) >= 1

;



SELECT first\_name, last\_name, department\_id

FROM Employee

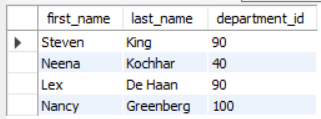
WHERE salary > (

SELECT AVG(salary)

FROM Employee

)

ORDER BY salary DESC;



* SELECT first\_name, last\_name, salary, department\_id

FROM Employee

WHERE salary > (

SELECT MAX(salary)

FROM Employee

WHERE department\_id = 40

);

