-- Create SALESPEOPLE table

CREATE TABLE SALESPEOPLE (

SNUM INT PRIMARY KEY,

SNAME VARCHAR(50),

CITY VARCHAR(50),

COMM DECIMAL(4,2)

);

-- Insert data into SALESPEOPLE

INSERT INTO SALESPEOPLE (SNUM, SNAME, CITY, COMM) VALUES

(1001, 'Peel', 'London', 0.12),

(1002, 'Serres', 'San Jose', 0.13),

(1004, 'Motika', 'London', 0.11),

(1007, 'Rafkin', 'Barcelona', 0.15),

(1003, 'Axelrod', 'New york', 0.10);

-- Create CUST table

CREATE TABLE CUST (

CNUM INT PRIMARY KEY,

CNAME VARCHAR(50),

CITY VARCHAR(50),

RATING INT,

SNUM INT,

FOREIGN KEY (SNUM) REFERENCES SALESPEOPLE(SNUM)

);

-- Insert data into CUST

INSERT INTO CUST (CNUM, CNAME, CITY, RATING, SNUM) VALUES

(2001, 'Hoffman', 'London', 100, 1001),

(2002, 'Giovanne', 'Rome', 200, 1003),

(2003, 'Liu', 'San Jose', 300, 1002),

(2004, 'Grass', 'Brelin', 100, 1002),

(2006, 'Clemens', 'London', 300, 1007),

(2007, 'Pereira', 'Rome', 100, 1004),

(2008, 'TempCustomer', 'Unknown', 100, 1007); -- Added for orders table

-- Create ORDERS table

CREATE TABLE ORDERS (

ONUM INT PRIMARY KEY,

AMT DECIMAL(10,2),

ODATE DATE,

CNUM INT,

SNUM INT,

FOREIGN KEY (CNUM) REFERENCES CUST(CNUM),

FOREIGN KEY (SNUM) REFERENCES SALESPEOPLE(SNUM)

);

-- Insert data into ORDERS

INSERT INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES

(3001, 18.69, '1994-10-03', 2008, 1007),

(3003, 767.19, '1994-10-03', 2001, 1001),

(3002, 1900.10, '1994-10-03', 2007, 1004),

(3005, 5160.45, '1994-10-03', 2003, 1002),

(3006, 1098.16, '1994-10-04', 2008, 1007),

(3009, 1713.23, '1994-10-04', 2002, 1003),

(3007, 75.75, '1994-10-05', 2004, 1002),

(3008, 4723.00, '1994-10-05', 2006, 1001),

(3010, 1309.95, '1994-10-06', 2004, 1002),

(3011, 9891.88, '1994-10-06', 2006, 1001);

/\*

select \* from SALESPEOPLE;

SELect \* from CUST;

select \* from ORDERS;

-- 1to5

SELECT snum,sname,city FROM SALESPEOPLE;

Select distinct snum from ORDERS;

select sname,COMM from SALESPEOPLE where city = 'London';

select CNAME from CUST where RATING = 100;

select onum,amt,odate from ORDERS;

-- 6to10

select CNAME from CUST where city = 'San Jose';

select CNAME from CUST where city = 'San Jose' and RATING > 200;

SELECT \* FROM orders WHERE amt > 1000;

SELECT sname, city FROM salespeople WHERE city = 'London' AND comm > 0.10;

SELECT \* FROM cust WHERE rating > 100 OR city = 'Rome';

-- 11to15

SELECT \* FROM salespeople WHERE city IN ('Barcelona', 'London');

SELECT \* FROM salespeople WHERE comm > 0.10 AND comm < 0.12;

SELECT \* FROM cust WHERE city IS NULL;

SELECT \* FROM orders WHERE odate IN ('1994-10-03', '1994-10-04');

SELECT \* FROM cust

WHERE snum IN (SELECT snum FROM salespeople WHERE sname IN ('Peel', 'Motika'));

-- 16to20

SELECT \* FROM cust WHERE cname LIKE 'A%' OR cname LIKE 'B%';

SELECT \* FROM orders WHERE amt IS NOT NULL AND amt <> 0;

SELECT COUNT(DISTINCT snum) FROM orders;

SELECT snum, odate, MAX(amt) AS max\_order

FROM orders

GROUP BY snum, odate;

SELECT snum, MAX(amt) AS max\_order

FROM orders

WHERE amt > 3000

GROUP BY snum;

-- 21to25

SELECT odate, SUM(amt) AS total\_amt

FROM orders

GROUP BY odate

ORDER BY total\_amt DESC

LIMIT 1;

SELECT COUNT(\*)

FROM orders

WHERE odate = '1994-10-03';

SELECT COUNT(DISTINCT city)

FROM cust

WHERE city IS NOT NULL;

SELECT cnum, MIN(amt) AS smallest\_order

FROM orders

GROUP BY cnum;

SELECT \*

FROM cust

WHERE cname LIKE 'G%'

ORDER BY cname ASC

LIMIT 1;

-- 26to30

SELECT

DATE\_FORMAT(odate, '%d/%m/%y') AS date,

COUNT(\*) AS order\_count,

CONCAT('For ', DATE\_FORMAT(odate, '%d/%m/%y'), ' there are ', COUNT(\*), ' orders.') AS result

FROM orders

GROUP BY odate;

SELECT onum, snum, amt \* 0.12 AS commission

FROM orders;

SELECT city, MAX(rating) AS highest\_rating

FROM cust

GROUP BY city;

SELECT odate, SUM(amt) AS total\_amt

FROM orders

GROUP BY odate

ORDER BY total\_amt DESC;

SELECT s.sname, c.cname, s.city

FROM salespeople s

JOIN cust c ON s.city = c.city;

-- 31to35

SELECT c.cname AS customer, s.sname AS salesperson

FROM cust c

JOIN salespeople s ON c.snum = s.snum;

SELECT o.onum, c.cname

FROM orders o

JOIN cust c ON o.cnum = c.cnum;

SELECT o.onum, s.sname AS salesperson, c.cname AS customer

FROM orders o

JOIN salespeople s ON o.snum = s.snum

JOIN cust c ON o.cnum = c.cnum;

SELECT c.cname

FROM cust c

JOIN salespeople s ON c.snum = s.snum

WHERE s.comm > 0.12;

SELECT o.onum, o.amt \* s.comm AS commission

FROM orders o

JOIN cust c ON o.cnum = c.cnum

JOIN salespeople s ON o.snum = s.snum

WHERE c.rating > 100;

-- 36-40

SELECT A.cname AS customer1, B.cname AS customer2, A.rating

FROM cust A, cust B

WHERE A.rating = B.rating AND A.cnum <> B.cnum;

SELECT A.cname AS customer1, B.cname AS customer2, A.rating

FROM cust A, cust B

WHERE A.rating = B.rating AND A.cnum < B.cnum;

SELECT c.cname, s.sname

FROM cust c

CROSS JOIN salespeople s;

SELECT cname FROM cust

WHERE city IN

(

SELECT city FROM cust WHERE snum =

(

SELECT snum FROM salespeople WHERE sname = 'Serres'

)

);

SELECT A.cname AS customer1, B.cname AS customer2, A.snum

FROM cust A, cust B

WHERE A.snum = B.snum AND A.cnum < B.cnum;

-- 41to45

SELECT A.sname AS salesperson1, B.sname AS salesperson2, A.city

FROM salespeople A, salespeople B

WHERE A.city = B.city AND A.snum < B.snum;

SELECT A.onum AS order1, B.onum AS order2, C.cname

FROM orders A

JOIN orders B ON A.cnum = B.cnum AND A.onum < B.onum

JOIN cust C ON A.cnum = C.cnum;

SELECT cname, city

FROM cust

WHERE rating = (

SELECT rating

FROM cust

WHERE cname = 'Hoffman'

);

SELECT o.\*

FROM orders o

JOIN salespeople s ON o.snum = s.snum

WHERE s.sname = 'Motika';

SELECT o.\*

FROM orders o

WHERE snum = (

SELECT snum

FROM cust

WHERE cname = 'Hoffman'

);

-- 45-50

SELECT o.\*

FROM orders o

WHERE snum = (

SELECT snum

FROM cust

WHERE cname = 'Hoffman'

);

SELECT \*

FROM orders

WHERE odate = '1994-10-04' AND amt > (

SELECT AVG(amt)

FROM orders

WHERE odate = '1994-10-04'

);

SELECT AVG(comm) AS avg\_commission

FROM salespeople

WHERE city = 'London';

SELECT o.\*

FROM orders o

WHERE snum IN (

SELECT snum

FROM cust

WHERE city = 'London'

);

SELECT DISTINCT s.comm

FROM salespeople s

JOIN cust c ON s.snum = c.snum

WHERE c.city = 'London';

SELECT \*

FROM cust

WHERE cnum = (

SELECT snum + 1000

FROM salespeople

WHERE sname = 'Serres'

);

-- 51. Count the customers with rating above San Jose’s average rating

SELECT COUNT(\*) AS count\_above\_avg\_rating

FROM cust

WHERE rating > (

SELECT AVG(rating)

FROM cust

WHERE city = 'San Jose'

);

-- 52. All orders for the customer named Cisnerous (assume unknown cnum)

SELECT o.\*

FROM orders o

JOIN cust c ON o.cnum = c.cnum

WHERE c.cname = 'Cisnerous';

-- 53. Names and rating of customers who have above average orders

SELECT DISTINCT c.cname, c.rating

FROM cust c

JOIN orders o ON c.cnum = o.cnum

GROUP BY c.cname, c.rating

HAVING AVG(o.amt) > (

SELECT AVG(amt) FROM orders

);

-- 54. Total order amount per salesperson, only if greater than max order amount in table

SELECT o.snum, SUM(o.amt) AS total\_amt

FROM orders o

GROUP BY o.snum

HAVING SUM(o.amt) > (

SELECT MAX(amt) FROM orders

);

-- 55. All customers with orders on 3rd Oct

SELECT DISTINCT c.\*

FROM cust c

JOIN orders o ON c.cnum = o.cnum

WHERE o.odate = '1994-10-03';

-- 56. Salespeople with more than one customer

SELECT s.snum, s.sname

FROM salespeople s

JOIN cust c ON s.snum = c.snum

GROUP BY s.snum, s.sname

HAVING COUNT(DISTINCT c.cnum) > 1;

-- 57. Check if correct salesperson was credited with each sale

SELECT o.\*, c.snum AS expected\_snum

FROM orders o

JOIN cust c ON o.cnum = c.cnum

WHERE o.snum <> c.snum;

-- 58. Orders with above average amount for their customer

SELECT o.\*

FROM orders o

JOIN (

SELECT cnum, AVG(amt) AS avg\_amt

FROM orders

GROUP BY cnum

) AS avg\_orders ON o.cnum = avg\_orders.cnum

WHERE o.amt > avg\_orders.avg\_amt;

-- 59. Sums of amounts per date, only if sum is at least 2000 above max order

SELECT odate, SUM(amt) AS total\_amt

FROM orders

GROUP BY odate

HAVING SUM(amt) >= (

SELECT MAX(amt) FROM orders

) + 2000;

-- 60. Customers with ratings equal to the maximum in their city

SELECT c1.\*

FROM cust c1

WHERE rating = (

SELECT MAX(c2.rating)

FROM cust c2

WHERE c2.city = c1.city

);

-- 61. Salespeople who have customers in their cities but do not service them (using JOIN)

SELECT DISTINCT s.snum, s.sname, s.city

FROM salespeople s

JOIN cust c ON s.city = c.city

WHERE s.snum <> c.snum;

-- 61 (Alternate): Using correlated subquery

SELECT s.snum, s.sname, s.city

FROM salespeople s

WHERE EXISTS (

SELECT 1

FROM cust c

WHERE c.city = s.city AND c.snum <> s.snum

);

-- 62. Customers in San Jose (only if at least one customer exists in San Jose)

SELECT cnum, cname, city

FROM cust

WHERE EXISTS (

SELECT 1

FROM cust

WHERE city = 'San Jose'

);

-- 63. Salespeople with multiple customers

SELECT snum

FROM cust

GROUP BY snum

HAVING COUNT(\*) > 1;

-- 64. Salespeople number, name, and city who have multiple customers

SELECT s.snum, s.sname, s.city

FROM salespeople s

JOIN cust c ON s.snum = c.snum

GROUP BY s.snum, s.sname, s.city

HAVING COUNT(\*) > 1;

-- 65. Salespeople who serve only one customer

SELECT s.snum, s.sname

FROM salespeople s

JOIN cust c ON s.snum = c.snum

GROUP BY s.snum, s.sname

HAVING COUNT(\*) = 1;

-- 66. Salespeople with more than one current order

SELECT s.snum, s.sname

FROM salespeople s

JOIN orders o ON s.snum = o.snum

GROUP BY s.snum, s.sname

HAVING COUNT(\*) > 1;

-- 67. Salespeople who have customers with rating of 300 (using EXISTS)

SELECT DISTINCT s.snum, s.sname

FROM salespeople s

WHERE EXISTS (

SELECT 1

FROM cust c

WHERE c.snum = s.snum AND c.rating = 300

);

-- 68. Salespeople who have customers with rating of 300 (using JOIN)

SELECT DISTINCT s.snum, s.sname

FROM salespeople s

JOIN cust c ON s.snum = c.snum

WHERE c.rating = 300;

-- 69. Salespeople with customers in their city but not assigned to them (using EXISTS)

SELECT s.snum, s.sname

FROM salespeople s

WHERE EXISTS (

SELECT 1

FROM cust c

WHERE c.city = s.city AND c.snum <> s.snum

);

-- 70. Customers assigned to salesperson who has at least one other customer with orders

SELECT DISTINCT c1.\*

FROM cust c1

WHERE EXISTS (

SELECT 1

FROM cust c2

JOIN orders o ON o.cnum = c2.cnum

WHERE c2.snum = c1.snum AND c2.cnum <> c1.cnum

);

-- 71. Salespeople with customers located in their cities (using ANY)

SELECT DISTINCT s.snum, s.sname

FROM salespeople s

WHERE s.city = ANY (SELECT city FROM cust WHERE cust.snum = s.snum);

-- 71 (alternate using IN)

SELECT DISTINCT s.snum, s.sname

FROM salespeople s

WHERE s.city IN (

SELECT city FROM cust WHERE cust.snum = s.snum

);

-- 72. Salespeople for whom customers follow them alphabetically (ANY + EXISTS)

SELECT s.snum, s.sname

FROM salespeople s

WHERE EXISTS (

SELECT 1 FROM cust c

WHERE c.snum = s.snum AND c.cname > s.sname

);

-- 73. Customers who have a greater rating than ANY customer in Rome

SELECT \*

FROM cust

WHERE rating > ANY (

SELECT rating FROM cust WHERE city = 'Rome'

);

-- 74. Orders with amounts greater than at least one order on Oct 6

SELECT \*

FROM orders

WHERE amt > ANY (

SELECT amt FROM orders WHERE odate = '1994-10-06'

);

-- 75. Orders with amounts smaller than any for customers in San Jose (using ANY)

SELECT \*

FROM orders

WHERE amt < ANY (

SELECT amt FROM orders o2

JOIN cust c ON o2.cnum = c.cnum

WHERE c.city = 'San Jose'

);

-- 75 (Alternate without ANY)

SELECT o.\*

FROM orders o

JOIN cust c1 ON o.cnum = c1.cnum

WHERE NOT EXISTS (

SELECT 1

FROM orders o2

JOIN cust c2 ON o2.cnum = c2.cnum

WHERE c2.city = 'San Jose' AND o.amt >= o2.amt

);

-- 76. Customers with ratings higher than every customer in Paris (using ALL)

SELECT \*

FROM cust

WHERE rating > ALL (

SELECT rating FROM cust WHERE city = 'Paris'

);

-- 76 (Alternate using NOT EXISTS)

SELECT \*

FROM cust c1

WHERE NOT EXISTS (

SELECT 1 FROM cust c2 WHERE c2.city = 'Paris' AND c1.rating <= c2.rating

);

-- 77. Customers whose ratings are equal to or greater than ANY of the customers served by Serres

SELECT \*

FROM cust

WHERE rating >= ANY (

SELECT rating FROM cust WHERE snum = (SELECT snum FROM salespeople WHERE sname = 'Serres')

);

-- 78. Salespeople with no customers in their city (using ANY)

SELECT \*

FROM salespeople s

WHERE s.city <> ALL (

SELECT city FROM cust WHERE cust.snum = s.snum

);

-- 78 (Alternate using ALL)

SELECT \*

FROM salespeople s

WHERE NOT EXISTS (

SELECT 1 FROM cust c WHERE c.city = s.city

);

-- 79. Orders greater than any for customers in London

SELECT \*

FROM orders

WHERE amt > ANY (

SELECT amt FROM orders o2

JOIN cust c ON o2.cnum = c.cnum

WHERE c.city = 'London'

);

-- 80. Salespeople and customers located in London

SELECT s.sname AS salesperson, c.cname AS customer

FROM salespeople s

JOIN cust c ON s.city = 'London' AND c.city = 'London';

\*/

-- 80. All salespeople and customers located in London

SELECT s.sname AS salesperson, c.cname AS customer

FROM salespeople s

JOIN cust c ON s.city = 'London' AND c.city = 'London';

-- 81. For every salesperson, dates on which highest and lowest orders were brought

SELECT o.snum,

MAX(o.amt) AS highest\_amt,

MIN(o.amt) AS lowest\_amt,

(SELECT o1.odate FROM orders o1 WHERE o1.snum = o.snum ORDER BY o1.amt DESC LIMIT 1) AS highest\_date,

(SELECT o2.odate FROM orders o2 WHERE o2.snum = o.snum ORDER BY o2.amt ASC LIMIT 1) AS lowest\_date

FROM orders o

GROUP BY o.snum;

-- 82. List all salespeople and indicate those who have customers in their cities and those who don’t

SELECT s.sname,

CASE

WHEN EXISTS (

SELECT 1 FROM cust c WHERE c.city = s.city AND c.snum = s.snum

) THEN 'Has Customer in City'

ELSE 'No Customer in City'

END AS status

FROM salespeople s;

-- 83. Append strings to selected fields indicating whether salesperson was matched to a customer in his city

SELECT s.sname,

s.city,

CONCAT(s.sname, ' - ',

CASE

WHEN EXISTS (

SELECT 1 FROM cust c WHERE c.city = s.city AND c.snum = s.snum

) THEN 'Matched'

ELSE 'Not Matched'

END) AS match\_status

FROM salespeople s;

-- 84. Union: customer name, city, rating with label 'High Rating' or 'Low Rating'

SELECT cname, city, rating, 'High Rating' AS status

FROM cust

WHERE rating >= 200

UNION

SELECT cname, city, rating, 'Low Rating' AS status

FROM cust

WHERE rating < 200;

-- 85. Name and number of each salesperson and customer with more than one order (alphabetical order)

SELECT s.sname AS name, s.snum AS number

FROM salespeople s

WHERE s.snum IN (

SELECT snum FROM orders GROUP BY snum HAVING COUNT(\*) > 1

)

UNION

SELECT c.cname, c.cnum

FROM cust c

WHERE c.cnum IN (

SELECT cnum FROM orders GROUP BY cnum HAVING COUNT(\*) > 1

)

ORDER BY name;

-- 86. Union of three queries: snums in San Jose, cnums in San Jose, onums on Oct 3

SELECT snum FROM salespeople WHERE city = 'San Jose'

UNION

SELECT cnum FROM cust WHERE city = 'San Jose'

UNION ALL

SELECT onum FROM orders WHERE odate = '1994-10-03';

-- 87. Salespeople in London who had at least one customer there

SELECT DISTINCT s.sname

FROM salespeople s

JOIN cust c ON s.snum = c.snum

WHERE s.city = 'London' AND c.city = 'London';

-- 88. Salespeople in London who did not have customers there

SELECT s.sname

FROM salespeople s

WHERE s.city = 'London' AND NOT EXISTS (

SELECT 1 FROM cust c WHERE c.snum = s.snum AND c.city = 'London'

);

-- 89. Salespeople matched to their customers, including unassigned salespeople (outer join + union)

SELECT s.sname, c.cname

FROM salespeople s

LEFT JOIN cust c ON s.snum = c.snum

UNION

SELECT s.sname, NULL

FROM salespeople s

WHERE s.snum NOT IN (SELECT DISTINCT snum FROM cust);