

SQL exercises, based on dreamhome database

Datatabase system:

1. Login to mysql database server imc.kean.edu using xxx account
ANS:
`mysql -u xxx -p -h imc.kean.edu`
2. Switch to database dreamhome
ANS:
`use dreamhome;`
3. Show all tables in dreamhome database
ANS:
`show tables;`
4. Show the table structure (column name, datatype, keys, etc) of Staff table
ANS:
`desc Staff;`

SELECT:

5. Show all the rows and all the columns of Staff table
ANS:
`SELECT * FROM Staff;`
6. Produce a list of salaries for all staff, showing only the staff number, the first and last names, and the salary details.
ANS:
`SELECT staffNo, fName, lName, salary FROM Staff;`
7. List full details of all hotels in London.
ANS:
`SELECT * FROM Hotel WHERE city = 'London';`
8. List the unique property numbers of all properties that have been viewed.
ANS:
`SELECT DISTINCT propertyNo FROM Viewing;`
9. Write a SQL query to list the staff who work in London. Please show their first name, last name and branchno.
ANS:
`SELECT fName,lName,b.branchno FROM Staff s, Branch b
WHERE s.branchno=b.branchno and b.city='London';`
10. Produce a list of monthly salaries for all staff, showing the staff number, the first and last names, and the salary details.
ANS:
`SELECT staffNo, fName, lName, salary/12 FROM Staff;`
11. List all staff with a salary greater than \$10,000.
ANS:
`SELECT staffNo, fName, lName, position, salary FROM Staff WHERE salary > 10000;`
12. Write a SQL query to display the name for vendor lives in New York with zipcode 07811.
(refer to A_Vendors)
ANS:
`SELECT name from A_Vendors where State = 'NY' and Zipcode='07811';`
13. Write a SQL query to list the unique position for female staff with salary less than 10000.
ANS:
`SELECT distinct position FROM Staff WHERE sex='F' and salary < 10000;`
14. List the addresses of all branch offices in London or Glasgow.
ANS:

`SELECT * FROM Branch WHERE city = 'London' OR city = 'Glasgow';`

15. Write a SQL query to display product names that make profit < 20. (refer to A_Products)

ANS:

`SELECT name FROM A_Products WHERE sell_price - cost < 20;`

16. List all staff with a salary between \$20,000 and \$30,000.

ANS:

`SELECT staffNo, fName, lName, position, salary
FROM Staff WHERE salary BETWEEN 20000 AND 30000;`

ANS 2:

`SELECT staffNo, fName, lName, position, salary
FROM Staff WHERE salary >= 20000 AND salary <= 30000;`

17. List all managers and supervisors.

ANS:

`SELECT staffNo, fName, lName, position FROM Staff WHERE position IN ('Manager', 'Supervisor');`

ANS 2:

`SELECT staffNo, fName, lName, position FROM Staff WHERE position = 'Manager' OR position = 'Supervisor';`

18. What is the output of the following SQL query?

`SELECT clientno, propertyno FROM Viewing WHERE comment!=""`

ANS:

clientno	propertyno
CR56	PA14
CR62	PA14
CR76	PG4

19. List all double or family rooms with a price below 40.00 per night, in ascending order of price.

ANS:

`SELECT * FROM Room WHERE price < 40 AND type IN ('double', 'family') ORDER BY price;`

20. Find all owners with the string 'Glasgow' in their address.

ANS:

`SELECT ownerNo, fName, lName, address, telNo
FROM PrivateOwner WHERE address LIKE '%Glasgow%';`

21. Write a SQL query to display the total number of Customers living in Union.

ANS:

`SELECT count(name) FROM A_Customers WHERE address like '%Union%';`

22. List the details of all viewings on property PG4 where a comment has not been supplied.

ANS:

`SELECT clientNo, viewDate FROM Viewing WHERE propertyNo = 'PG4' AND comment IS NULL;`

23. Produce a list of salaries for all staff, arranged in descending order of salary.

ANS:

`SELECT staffNo, fName, lName, salary FROM Staff ORDER BY salary DESC;`

24. Produce an abbreviated list of properties arranged in order of property type.

ANS:

`SELECT propertyNo, type, rooms, rent FROM PropertyForRent ORDER BY type;`

25. List the names and addresses of all guests in London, alphabetically ordered by name.

ANS:

`SELECT guestName, guestAddress FROM Guest
WHERE guestAddress LIKE '%London%' ORDER BY guestName;`

AGGREGATE FUNCTIONS, GROUP BY:

26. Write a SQL query to display the number of unique products been reviewed (A_Reviews table).

ANS:

SELECT count(distinct p_id) FROM A_Reviews;

27. Write a SQL query to display the total salary of all staff.

ANS:

SELECT SUM(salary) from Staff;

28. What is the output of the following SQL query?

SELECT max(price) as myMax FROM Room WHERE type='family';

ANS:

myMax
59.99

29. What is the output of the following SQL query?

SELECT type, count(*) as ct,max(price) FROM Room group by type having ct > 3;

ANS:

type	ct	max(price)
double	4	86.00
single	5	58.00

30. Write a SQL query to list the unique branchno that branch has at least two female staff.

ANS:

SELECT distinct branchno FROM Staff group by branchno,sex having count(sex) >= 2 and sex='F';

31. Write a SQL query to display the total revenues of all products. The output column name should be "revenue".

ANS:

SELECT sum(sell_price*quantity) as revenue FROM A_Products;

32. Write the SQL query to show the branchno and the maximum salary for each branch in the Staff table. The output should be sorted from low to high based on the max salary of each branch;

ANS:

SELECT branchno, max(salary) FROM Staff group by branchno order by max(salary) desc;

33. Can we run the following SQL query without any problem?

SELECT fname FROM Staff WHERE salary > max(salary);

ANS:

NO, the statement invalid use of group function. Aggregation cannot be in WHERE.

34. Write a SQL query to find the staff name (first name, last name) who has salary higher than the average salary of branchno ='B003';

ANS:

SELECT fname,lname FROM Staff where salary > (select avg(salary) from Staff WHERE branchno ='B003');

35. Write the SQL query to find how many properties cost more than \$350 per month to rent?

ANS:

SELECT COUNT(*) AS myCount FROM PropertyForRent WHERE rent > 350;

36. How many different properties were viewed in May 2013?

ANS:

SELECT COUNT(DISTINCT propertyNo) AS myCount FROM Viewing WHERE viewDate BETWEEN '1-May-13' AND '31-May-13';

37. Find the total number of Managers and the sum of their salaries.

ANS:

```
SELECT COUNT(staffNo) AS myCount, SUM(salary) AS mySum FROM Staff WHERE position = 'Manager';
```

38. Find the minimum, maximum, and average staff salary.

ANS:

```
SELECT MIN(salary) AS myMin, MAX(salary) AS myMax, AVG(salary) AS myAvg FROM Staff;
```

39. What is the total **revenue** per night from all **double** type rooms if all rooms are booked?

ANS:

```
SELECT SUM(price) FROM Room WHERE type = 'double';
```

40. Find the number of staff working in each branch and the sum of their salaries.

ANS:

```
SELECT branchNo, COUNT(staffNo) AS myCount, SUM(salary) AS mySum FROM Staff GROUP BY branchNo ORDER BY branchNo;
```

ANS 2:

```
SELECT branchNo, ( SELECT COUNT(staffNo) AS myCount FROM Staff s WHERE s.branchNo = b.branchNo ), ( SELECT SUM(salary) AS mySum FROM Staff s WHERE s.branchNo = b.branchNo )FROM Branch b ORDER BY branchNo;
```

41. For each branch office with more than one member of staff, find the number of staff working in each branch and the sum of their salaries.

ANS:

```
SELECT branchNo, COUNT(staffNo) AS myCount, SUM(salary) AS mySum FROM Staff GROUP BY branchNo HAVING COUNT(staffNo) > 1 ORDER BY branchNo;
```

42. List the hotel name and the number of rooms in each hotel.

ANS:

```
SELECT h.hotelname, COUNT(r.roomNo) AS count FROM Room r, Hotel h where r.hotelNo=h.hotelNo GROUP BY h.hotelNo;
```

43. What is the output of the following SQL statement?

```
SELECT count(comment) FROM Viewing;
```

ANS:

```
+-----+
| count(comment) |
+-----+
|                |
|                |
|                |
|                |
+-----+
```

44. Show the total revenue per night for the double type room if all rooms are booked.

ANS:

```
SELECT SUM(price) FROM Room WHERE type = 'double';
```

SUBQUERY:

45. Can we run the following query without any problem?

```
SELECT city FROM Branch WHERE branchNo = (SELECT branchNo FROM Staff where sex='F');
```

ANS:

Error, Subquery returns more than 1 row. Cannot use =, should use IN

46. What is the output of the following SQL query?

```
SELECT c.name FROM A_Customers c WHERE c.id NOT in (SELECT c_id FROM A_Reviews);
```

ANS:

```
+-----+
| name |
+-----+
| BJ2  |
| BJ3  |
| Judy1|
| Judy2|
+-----+
```

47. What is the output of the following SQL query?

```
SELECT count(distinct(city)) as myCt FROM Hotel WHERE hotelno in (SELECT hotelno FROM Room WHERE type='single');
```

ANS:

```
+-----+
| myCt |
+-----+
| 3    |
+-----+
```

48. Write a SQL query to display the staffno and working branch city for the staff who has the lowest salary.

ANS:

```
SELECT s.staffno,b.city FROM Staff s, Branch b
WHERE s.branchno=b.branchno and s.salary in (SELECT min(salary) FROM Staff);
```

49. List the staff who work in the branch at '163 Main St'.

ANS:

```
SELECT staffNo, fName, IName, position FROM Staff
WHERE branchNo = (SELECT branchNo FROM Branch WHERE street = '163 Main St');
```

ANS 2:

```
SELECT staffNo, fName, IName, position FROM Staff s, Branch b
WHERE s.branchNo = b.branchNo and street = '163 Main St';
```

50. List all guests staying at the Grosvenor Hotel from April 1st to August 31st, 2004.

ANS:

```
SELECT * FROM Guest WHERE guestNo in (
    SELECT guestNo FROM Booking
    WHERE dateFrom >= date('2004-04-01') AND dateFrom <= date('2004-08-31')
    AND hotelNo = (SELECT hotelNo FROM Hotel WHERE hotelName = 'Grosvenor'));
```

51. List all staff whose salary is greater than the average salary, and show by how much their salary is greater than the average.

ANS:

```
SELECT staffNo, fName, LName, position, (salary - (SELECT AVG(salary) FROM Staff)) AS salDiff FROM
Staff WHERE salary > (SELECT AVG(salary) FROM Staff);
```

ANS 2:

```
SELECT staffNo, fName, LName, position, (salary-a.avg1) AS salDiff
FROM Staff s, (SELECT AVG(salary) as avg1 FROM Staff) a WHERE salary > a.avg1;
```

52. List the properties that are handled by staff who work in the branch at '163 Main St'.

ANS:

```
SELECT propertyNo, street, city, postcode, type, rooms, rent
FROM PropertyForRent WHERE staffNo IN (SELECT staffNo
FROM Staff WHERE branchNo = (SELECT branchNo
FROM Branch WHERE street = '163 Main St'));
```

53. Find all staff whose salary is larger than the salary of at least one member of staff at branch B003.

ANS:

```
SELECT staffNo, fName, IName, position, salary
FROM Staff WHERE salary > SOME (SELECT salary FROM Staff WHERE branchNo = 'B003');
```

54. Find all staff whose salary is larger than the salary of every member of staff at branch B003.

ANS:

```
SELECT staffNo, fName, IName, position, salary
FROM Staff WHERE salary > ALL (SELECT salary FROM Staff WHERE branchNo = 'B003');
```

55. Write a SQL query to show client names who didn't give reviews (Viewing table).

ANS:

```
SELECT c.fname,c.lname FROM Client c
```

WHERE c.clientno NOT in (SELECT clientno FROM Viewing);

56. What is the most commonly booked room type for each hotel in London?

ANS:

```
SELECT t.hotelname, t.type, MAX(t.X)
FROM ( SELECT hotelname ,type, COUNT(type) AS X
      FROM Booking b, Hotel h, Room r
      WHERE r.roomNo = b.roomNo AND b.hotelNo = h.hotelNo AND h.city = 'London'
      GROUP BY h.hotelname, r.type) t
GROUP BY t.hotelname;
```

57. Write a SQL query to display the cheapest product name (A_Products table) and its price.

ANS:

```
SELECT name, sell_price FROM A_Products
WHERE sell_price = (SELECT min(sell_price) FROM A_Products);
```

(INNER) JOIN

58. How many rows and columns will be generated from the following SQL query?

```
SELECT * FROM A_Students, A_Courses;
```

ANS:

6 columns and 8 rows

59. Write a SQL statement to show the product names provided by vendor "James".

(refer to A_Products, A_Vendors tables)

ANS:

```
SELECT p.name FROM A_Products p, A_Vendors v WHERE v.name='James' and p.V_Id=v.V_Id;
```

60. Write a SQL query to show the first name who works at London and has salary higher than 10000.

ANS:

```
SELECT fname FROM Branch b, Staff s
WHERE b.branchno=s.branchno and b.city='London' and s.salary > 10000;
```

61. List the names of all clients who have viewed a property, along with any comments supplied.

ANS:

```
SELECT c.clientNo, fName, IName, propertyNo, comment
FROM Client c, Viewing v WHERE c.clientNo = v.clientNo;
```

62. Write a SQL query to list employee names (first name, last name) and their Supervisor names (first name, last name). The output header must clearly indicates who is employee and Supervisor. (refer to A_Employee table)

ANS:

```
SELECT e.fname as E_fname, e.lname as E_lname, m.fname as M_fname, m.lname as M_lname
FROM A_Employee e, A_Employee m WHERE e.super_ssn=m.ssn;
```

63. For each branch office, list the staff numbers and names of staff who manage properties and the properties that they manage.

ANS:

```
SELECT s.branchNo, s.staffNo, fName, IName, propertyNo
FROM Staff s, PropertyForRent p WHERE s.staffNo = p.staffNo
ORDER BY s.branchNo, s.staffNo, propertyNo;
```

64. For each branch, list the staff numbers and names of staff who manage properties, including the city in which the branch is located and the properties that the staff manage.

ANS:

```
SELECT b.branchNo, b.city, s.staffNo, fName, IName, propertyNo
FROM Branch b, Staff s, PropertyForRent p
WHERE b.branchNo = s.branchNo AND s.staffNo = p.staffNo
ORDER BY b.branchNo, s.staffNo, propertyNo;
```

65. Display the product names and the comments that customer BJ1 gave.
(refer to A_Cusomters, A_Reviews, A_Products)

ANS:

```
SELECT p.name, comments FROM A_Customers c, A_Reviews r, A_Products p
WHERE c.id=r.c_id and p.id=r.p_id and c.name='BJ1' ;
```

66. Find the number of properties handled by each staff member, along with the branch number of the member of staff.

ANS:

```
SELECT s.branchNo, s.staffNo, COUNT(*) AS myCount
FROM Staff s, PropertyForRent p WHERE s.staffNo = p.staffNo
GROUP BY s.branchNo, s.staffNo ORDER BY s.branchNo, s.staffNo;
```

67. List all branch offices and any properties that are in the same city.

ANS:

```
SELECT b.*, p.* FROM Branch1 b LEFT JOIN PropertyForRent1 p ON b.bCity = p.pCity;
```

68. List all properties and any branch offices that are in the same city.

ANS:

```
SELECT b.*, p.* FROM Branch1 b RIGHT JOIN PropertyForRent1 p ON b.bCity = p.pCity;
```

69. List the branch offices and properties that are in the same city along with any unmatched branches or properties.

ANS:

```
SELECT b.*, p.* FROM Branch1 b FULL JOIN PropertyForRent1 p ON b.bCity = p.pCity;
```

70. Find all staff who work in a London branch office.

ANS:

```
SELECT staffNo, fName, lName, position
FROM Staff s WHERE EXISTS (SELECT * FROM Branch b
WHERE s.branchNo = b.branchNo AND city = 'London');
```

ANS 2:

```
SELECT staffNo, fName, lName, position
FROM Staff s, Branch b WHERE s.branFchNo = b.branchNo AND city = 'London';
```

OUTER JOIN:

71. How many rows will be generated from the following SQL statement:

```
SELECT * FROM Staff s RIGHT JOIN Branch b ON s.branchno=b.branchno;
```

ANS:

8 rows

72. Write a SQL query to display **all customer names** (A_Customers) and their number of Reviews (A_Reviews). If a customer didn't give review, his/her number of review will be 0. The output should have customer name in alphabetic order from A to Z.

ANS:

```
SELECT c.name, count(r.c_id) FROM A_Customers c
LEFT JOIN A_Reviews r on r.c_id= c.id group by c.id order by c.name ASC;
```

SET (UNION, INTERSECT, EXCEPT):

73. Construct a list of all cities where there is either a branch office or a property.

ANS:

```
(SELECT city FROM Branch WHERE city IS NOT NULL)
UNION (SELECT city FROM PropertyForRent WHERE city IS NOT NULL);
```

74. Construct a list of all cities where there is both a branch office and a property.

ANS 1:

```
(SELECT city FROM Branch) INTERSECT (SELECT city FROM PropertyForRent);
```

ANS 2:

```
SELECT DISTINCT b.city FROM Branch b,PropertyForRent p WHERE b.city=p.city;
```

75. Write a SQL query to find the vendor names not providing any products.
(refer to A_Vendors, A_Products tables)

ANS:

```
SELECT name FROM A_Vendors where V_Id not in (SELECT V_Id from A_Products);
```

76. Construct a list of all cities where there is a branch office but no properties.

ANS:

```
(SELECT city FROM Branch) EXCEPT (SELECT city FROM PropertyForRent);
```

ANS 2:

```
SELECT DISTINCT city FROM Branch WHERE city NOT IN (SELECT city FROM PropertyForRent);
```

INSERT:

77. Insert a new row into the Staff table supplying data for all columns.

ANS:

```
INSERT INTO Staff VALUES
```

```
('SG16', 'Alan', 'Brown', 'Assistant', 'M', DATE '1957-05-25', 8300, 'B003');
```

78. Insert a new row into the Staff table supplying data for all mandatory columns: staffNo, fName, lName, position, salary, and branchNo.

ANS:

```
INSERT INTO Staff (staffNo, fName, lName, position, salary, branchNo)
```

```
VALUES ('SG44', 'Anne', 'Jones', 'Assistant', 8100, 'B003');
```

ANS 2:

```
INSERT INTO Staff VALUES ('SG44', 'Anne', 'Jones', 'Assistant', NULL, NULL, 8100, 'B003');
```

79. Assume that there is a table StaffPropCount that contains the names of staff and the number of properties they manage: StaffPropCount(staffNo, fName, lName, propCount)
Write a SQL to populate the StaffPropCount table using details from the Staff and PropertyForRent tables.

ANS:

```
INSERT INTO StaffPropCount (SELECT s.staffNo, fName, lName, COUNT(*)
```

```
FROM Staff s, PropertyForRent p
```

```
WHERE s.staffNo = p.staffNo GROUP BY s.staffNo, fName, lName)
```

```
UNION
```

```
(SELECT staffNo, fName, lName, 0 FROM Staff s
```

```
WHERE NOT EXISTS (SELECT * FROM PropertyForRent p WHERE p.staffNo = s.staffNo));
```

UPDATE:

80. Give all staff a 3% pay increase.

ANS:

```
UPDATE Staff SET salary = salary*1.03;
```

81. Give all Managers a 5% pay increase. (Refer to Staff table)

ANS:

```
UPDATE Staff SET salary = salary*1.05 WHERE position = 'Manager';
```

82. Promote David Ford (staffNo = 'SG14') to Manager and change his salary to \$18,000.
(Refer to Staff table)

ANS:

```
UPDATE Staff SET position = 'Manager', salary = 18000 WHERE staffNo = 'SG14';
```

DELETE:

83. Delete all viewings that relate to property PG4.

ANS:

DELETE FROM Viewing WHERE propertyNo = 'PG4';

84. Delete all rows from the Viewing table.

ANS:

DELETE FROM Viewing;

85. Delete the record that has the NULL value in QTY field in the A_Sales table

ANS:

DELETE FROM A_Sales WHERE QTY is null;

86. Create the Staff table with proper data type for each field and staffno is the primary key, branchno is a foreign key reference to the primary key branchno in the Branch table. All the fields cannot be NULL.

ANS:

CREATE TABLE Staff (staffno varchar(5) not null, fName varchar(15) not null, lName varchar(15) not null, position varchar(25) not null, Sex char(1) not null, DOB date not null, salary decimal(8,2) not null, branchno varchar(8) not null, primary key (staffno), foreign key (branchno) references Branch(branchno));

ALTER:

87. Add a new column TEL varchar(20) to the staff table;

ANS:

ALTER TABLE Staff ADD TEL varchar(25);

88. Change the Staff table by removing the default of 'Assistant' for the position column and setting the default for the sex column to female ('F').

ANS:

ALTER TABLE Staff ALTER position DROP DEFAULT;

ALTER TABLE Staff ALTER sex SET DEFAULT 'F';

VIEW:

89. Create a view vTest listing all double or family rooms with a price below 40.00 per night, in ascending order of price.

ANS:

**CREATE VIEW vTest as SELECT * FROM Room
WHERE price < 40 AND type IN ('double', 'family') ORDER BY price asc;**

90. Create a view vTest to list the number of rooms in each hotel.

ANS:

**SELECT h.hotelname, COUNT(r.roomNo) AS count
FROM Room r, Hotel h where r.hotelNo=h.hotelNo GROUP BY h.hotelNo;**

91. Create a view vTest to list the hotel name and the total number of rooms with price per room below 40.00 per night.

ANS:

**CREATE VIEW vTest as SELECT h.hotelname,count(*) FROM Hotel h, Room r
WHERE h.hotelNo=r.hotelNo and r.price < 40 group by h.hotelNo;**

92. Create a view so that the manager at branch B003 can see the details only for staff who work in his or her branch office.

ANS:

CREATE VIEW Manager3Staff AS SELECT * FROM Staff WHERE branchNo = 'B003';

93. Create a view of the staff details at branch B003 that excludes salary information, so that only managers can access the salary details for staff who work at their branch.

ANS:

```
CREATE VIEW Staff3 AS SELECT staffNo, fName, IName, position, sex  
FROM Staff WHERE branchNo = 'B003';
```

94. Create a view vTest to show the first name and the working city for staff who was born after January first 1960.

ANS:

```
SELECT fname, city FROM Staff s, Branch b  
WHERE s.branchno=b.branchno and birthday > '1960-01-01';
```

95. Create a view vTest to find the total revenue per night for each hotel and the revenue is higher than 85 dollars. Assume all rooms are booked. Your output needs to show the hotel name and the amount of revenue.

ANS:

```
SELECT h.hotelNo, SUM(price) as revenue FROM Hotel h, Room r  
WHERE h.hotelNo=r.hotelNo group by h.hotelNo having revenue > 85;
```

96. Create a view of staff who manage properties for rent, which includes the branch number they work at, their staff number, and the number of properties they manage

ANS:

```
CREATE VIEW StaffPropCnt (branchNo, staffNo, cnt)  
AS SELECT s.branchNo, s.staffNo, COUNT(*)  
FROM Staff s, PropertyForRent p WHERE s.staffNo = p.staffNo GROUP BY s.branchNo, s.staffNo;
```

GRANT, REVOKE:

97. Give the user with authorization identifier Manager all privileges on the Staff table.

ANS:

```
GRANT ALL PRIVILEGES ON Staff TO Manager WITH GRANT OPTION;
```

98. Give users Personnel and Director the privileges SELECT and UPDATE on column salary of the Staff table.

ANS:

```
GRANT SELECT, UPDATE (salary) ON Staff TO Personnel, Director;
```

99. Give user xyz from IP '10.20.30.40' the privileges SELECT on column salary of the Staff table.

ANS:

```
GRANT SELECT (s1) ON test TO 'xyz'@'10.20.30.40';
```

100. Remove user xyz's SELECT privilege from test the test table.

ANS:

```
REVOKE SELECT on test from xyz;
```

101. Remove user xyz's all privileges from test the test table.

ANS:

```
REVOKE ALL PRIVILEGES on test from xyz;
```

DROP:

102. Remove a table Data from the database.

ANS:

```
DROP TABLE table Data;
```

103. Remove a view vData from the database.

ANS:

```
DROP VIEW vData;
```

104. Remove a stored procedure pTest() from the database.

ANS:

```
DROP procedure pTest;
```

105. Remove a stored function fTest() from the database.

ANS:

DROP function fTest;

106. Remove a user xyz from the database.

ANS:

DROP USER xyz;

SQL Syntax and runtime issues:

107. Can the following SQL be run without error? SELECT staffNo, COUNT(salary) FROM Staff;

ANS:

No, this query is illegal because the query does not have a GROUP BY clause and the column staffNo in the SELECT list is used outside an aggregate function.

108. Is the following view updatable? CREATE VIEW vTest AS SELECT count(*) as ct FROM Staff;

ANS:

No, because the view is created with an aggregate function count()

109. Is the following view updatable? CREATE VIEW vTest AS SELECT fname, lname as ct FROM Staff;

ANS:

Yes, because every record in vTest can be traced back to the original record in Staff.

110. Is the following view updatable? CREATE VIEW vTest AS SELECT distinct branchno FROM Staff;

ANS:

No, because the view is created with the distinct keyword.

111. Is the following view updatable?

CREATE VIEW vTest AS SELECT sex FROM Staff GROUP BY sex;

ANS:

No, because the view is created with GROUP BY.

112. Is the following view updatable?

CREATE VIEW vTest AS SELECT branchno, sum(salary) FROM Staff GROUP BY branchno;

ANS:

No, because the view is created with an aggregate function sum() and GROUP BY.

113. Is the following view updatable?

CREATE VIEW vTest AS SELECT staffno, salary from Staff order by salary;

ANS:

Yes, because every record in vTest can be traced back to the original record in Staff. ORDER BY does not affect the trace.

VARIABLE, ASSIGNMENTS, COMPARISON:

114. What is the output for the following SQL:

SELECT 2=2, 3=2, 1=NULL, NULL=NULL, NULL IS NULL;

ANS:

2=2	3=2	1=NULL	NULL=NULL	NULL IS NULL
1	0	NULL	NULL	1

115. What is the output for the following SQL:

SELECT @a1:=3, @a2=4, @a3=@a2+1;

ANS:

@a1:=3	@a2=4	@a3=@a2+1
3	NULL	NULL

116. What is the output for the following SQL:

```
SELECT @a1:=3, @a2:=4, @a3:=@a2+@a1;
```

ANS:

@a1:=3	@a2:=4	@a3:=@a2+@a1
3	4	7

117. What is the output for the following SQL:

```
SELECT null and 1, null and 0, 1 and 0, null or 1, null or 0, 1 or 0, not null;
```

null and 1	null and 0	1 and 0	null or 1	null or 0	1 or 0	not null
NULL	0	0	1	NULL	1	NULL

STORED ROUTINES:

118. Write the SQL statement to run a stored function named fMaxSalary() which will return the staff's highest salary.

ANS:

```
SELECT fMaxSalary();
```

119. Write a stored function named fMaxSalary() which will return the staff's highest salary.

ANS:

```
DELIMITER $$
```

```
CREATE FUNCTION fMaxSalary() RETURNS float
```

```
BEGIN
```

```
    DECLARE maxSalary float ;
```

```
    SELECT max(salary) into maxSalary FROM dreamhome.Staff;
```

```
    RETURN maxSalary ;
```

```
END $$
```

```
DELIMITER ;
```

120. Write a stored function named fMaxSalary(bno) which will return the staff's highest salary for a given branch no bno.

ANS:

```
DELIMITER $$
```

```
CREATE FUNCTION fMaxSalary(bno varchar(10)) RETURNS float
```

```
BEGIN
```

```
    DECLARE maxSalary float ;
```

```
    SELECT max(salary) into maxSalary FROM dreamhome.Staff
```

```
    WHERE branchno=bno;
```

```
    RETURN maxSalary ;
```

```
END $$
```

```
DELIMITER ;
```

121. Write a stored function named fMaxSalary(n) which will return the staff's highest salary for a given input that is a partial string in the first name field.

ANS:

```
DELIMITER $$
```

```
CREATE FUNCTION fMaxSalary(n varchar(10)) RETURNS float
```

```
BEGIN
```

```
    DECLARE maxSalary float ;
```

```
    SELECT max(salary) into maxSalary FROM dreamhome.Staff
```

```
    WHERE fname like concat('%',n,'%');
```

```
    RETURN maxSalary ;
```

```
END $$
```

DELIMITER ;

122. Write the SQL statement to run a stored procedure named **pGetNames()** which will display the Staff's first name and last name who work at branch located at a given city 'London'.

ANS: **call pGetNames('London');**

123. Write a stored procedure named **pGetNames (bcity)** which will display the Staff's first name and last name who work at branch located at a given city bcity.

ANS:

DELIMITER \$\$

CREATE PROCEDURE pGetNames

(IN bcity VARCHAR(12))

BEGIN

SELECT fname,lname FROM dreamhome.Staff s, dreamhome.Branch b

WHERE city= bcity and b.branchno=s.branchno;

END \$\$

DELIMITER ;

124. Write a stored procedure named **pGetNames (bcity)** which will display the Staff's first name and last name who work at branch located at a given city bcity. If the bcity is empty "", please print a message saying "The input city cannot be empty."

call pGetNames('');	call pGetNames('London');
+-----+-----+ message +-----+-----+	+-----+-----+ fname lname +-----+-----+
Input city cannot be empty. +-----+-----+	John white Julie Lee +-----+-----+

ANS:

DELIMITER \$\$

CREATE PROCEDURE pGetNames

(IN bcity VARCHAR(12))

BEGIN

if (bcity='') THEN

SELECT 'Input city cannot be empty.' as message;

ELSE

SELECT fname,lname FROM dreamhome.Staff s, dreamhome.Branch b

WHERE city= bcity and b.branchno=s.branchno;

END IF;

END \$\$

DELIMITER ;