Writing basic select statements

The ***SELECT*** statement which represents the Data Query language in SQL is used to fetch data from one or more tables. It can consist of the list of columns to be fetched, or some additional clauses which specify what and how the data should be returned. It is used in conjunction with the FROM clause to extract data from the table.

Capabilities of select statement.

Projection: You can use the projection capability in SQL to choose the columns in a table that you want returned by your query. You can choose as few or as many columns of the table as you require.

Selection: You can use the selection capability in SQL to choose the rows in a table that you want returned by a query. You can use various criteria to selectively restrict the rows that you see.

Join: You can use the join capability in SQL to bring together data that is stored in different tables by creating a link through a column that both the tables share. You will learn more about joins in a later lesson.

Projection: Project specific column from the table

syntax:

```
SELECT  * | DISTINCT  | column1,column2,Expression, function ……..
FROM table_name;
```

Projecting all columns

Use ***** to project all columns from the table

Syntax:

```
SELECT * FROM table_name;
```

Example:
Find the details of all employees from the EMP table.

```
SELECT * FROM emp;
```

Selecting specific columns

List the names of the columns after select

Syntax :

```
SELECT column1,column2,..... FROM table_name;
```

Example :
Display the employee number,employee name , salary and department number of all employees.

```
SELECT empno, ename, sal, deptno FROM emp;
```

Note that the select statements are not case sensitive.

Eliminating the duplicate records

Use DISTINCT keyword

Syntax:

```
SELECT DISTINCT column_name,column_name….  FROM table_name;
```

Display the jobs available in the emp table.

```
SELECT DISTINCT job FROM emp;
```

Note : Only one distinct can be given for a select statement.

Sorting the output:

The order of rows that are returned by the query is undefined. To sort the output use ORDER BY clause. The ORDER BY clause must be the last clause in the select statement. The default mode of sorting is ascending order. For descending order use DESC.

Syntax:

```
SELECT column_list FROM table_name
WHERE condition
ORDER BY column_name/s,| columnAlias | column_position ;
```

Examples:

```
SELECT * FROM emp ORDER BY sal;
```

```
SELECT * FROM emp WHERE deptno=20 ORDER BY hiredate DESC;
```

```
SELECT * FROM emp ORDER BY deptno ,sal DESC;
```

```
SELECT * FROM emp ORDER BY 8 DESC;
```

Pagination:

Pagination is a technique used to divide large result sets into smaller pieces or pages.

Pagination is implemented using *LIMIT* and *OFFSET* clauses.

LIMIT clause is used to specify the maximum number of rows to return.

OFFSET clause is used to skip a certain number of rows before starting to return to rows.These two clauses must be the last clause after order by clause.

syntax:

```
SELECT column_list from table_name
Where condition
Order by column_name
Limit n  offset n;
```

Example:

Display the details of first 2 salesman from emp table

```
SELECT * FROM emp
WHERE job =  'SALESMAN'
LIMIT 2,
```

Example:

Display the details of second least paid manager

```
SELECT * FROM emp
WHERE job =  'MANAGER'
ORDER BY  sal
LIMIT 1 offset 1;
```

Column Alias

It is a temporary name given to a column or an expression in a select statement and used to improve readability particularly when performing calculations.

Use as keyword which is optional to specify the column alias. Enclose within quotes if the alias contains space

syntax:

```
SELECT column_name  as alias , expression  as "alias name" from table name
```

Example :

```
SELECT ename as Name , sal*12 as "Annual salary" from emp;
```

Selection :

The selection capability is used to filter the output based on conditions

The WHERE clause allows you to filter the output. It directly follows the FROM clause and if the condition is true, then the row is processed.

Syntax:

```
SELECT column_list FROM table_name WHERE condition ;
```

Examples:

```
SELECT * FROM emp WHERE ename='SCOTT';
SELECT ename,job,sal FROM emp WHERE sal>=1500;
```

Operators:

An operator manipulates individual data items and returns a result. The data items are called operands or arguments. These arguments can be user defined values or column names. Operators can be classified as

a) Arithmetic operators.
b) comparison or relational operators.
c) Logical operators.

A) Arithmetic operators.

Just as the name implies these operators are used to perform arithmetic computations.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division

Examples:

Calculate the salary of all employees after a raise of 100.

```
SELECT ename,sal,sal+100 FROM emp;
```

Display the annual salary of all employees after an increase of 1000

```
SELECT ename,sal,(sal+1000)*12 FROM emp;
```

Note

The order of precedence of the arithmetic operators

Parenthesis ()
Multiplication *
Division /
Addition + ,
subtraction -

B) Comparison operators

These are used in conditions that compare one expression with another.

Operator	Description	Example
=	Equality test	***SELECT * FROM emp WHERE ename='SCOTT';***
!= or <>	Inequality test	***SELECT ename,sal FROM emp WHERE job!='MANAGER'';***
>	Greater than	***SELECT * FROM emp WHERE sal>1600;***
>=	Greater than or equal to	***SELECT * FROM emp WHERE SAL>=2000;***

<	Less than	***SELECT * FROM emp WHERE sal<1600;***
<=	Less than or equal to	***SELECT * FROM emp WHERE SAL<=2000;***
[NOT] IN	Matching records in the given set of values	***SELECT * FROM emp WHERE deptno IN (10,20,30);***
[NOT]BETWEEN … AND	Select the rows within the specified range	***SELECT * FROM emp WHERE sal BETWEEN 1000 AND 2000;***
[NOT] LIKE	Select the rows if a character string matches the specified pattern	***SELECT ename FROM emp WHERE ename LIKE 'A%';***
IS [NOT] NULL	Tests for null.	***SELECT * FROM emp WHERE comm IS NULL;***

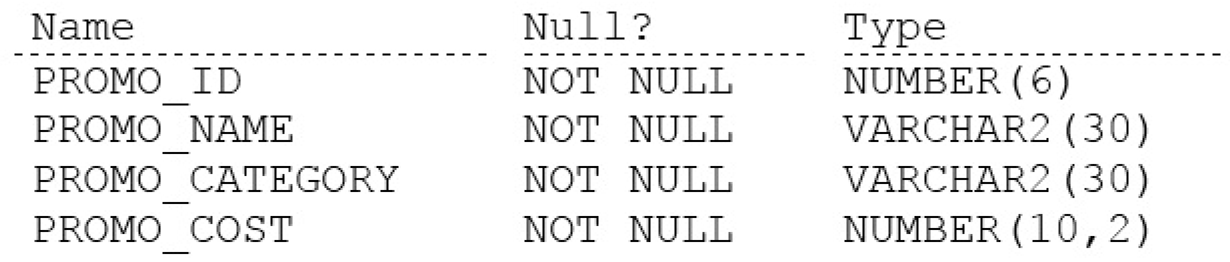
C) Logical Operators:

These operators compare two or more conditions at a time to determine whether a row can be processed or not. In other words these operators allow us to combine two or more conditions.

Operator	Description	Example
AND	select the row if both the conditions are true	***SELECT ename,job,sal FROM emp WHERE job='CLERK' AND sal <1200;***
OR	select the row if either one conditions is true	***SELECT * FROM emp WHERE deptno=10 OR deptno=20;***
NOT	select the row if conditions is false	***SELECT * FROM emp WHERE deptno NOT IN (10,20)***

Choose the correct choice to answer the following :

1. Examine the description of the PROMOTIONS table:



```
 Name                           Null?            Type
 ------------------------------ ---------------- ------------------------
 PROMO_ID                       NOT NULL         NUMBER(6)
 PROMO_NAME                     NOT NULL         VARCHAR2(30)
 PROMO_CATEGORY                 NOT NULL         VARCHAR2(30)
 PROMO_COST                     NOT NULL         NUMBER(10,2)
```

   You want to display the unique promotion costs in each promotion category.
   Which two queries can be used? (Choose two.)

   A. SELECT DISTINCT promo_category,promo_cost FROM promotions ORDER BY promo_category;

   B. SELECT DISTINCT promo_cost , DISTINCT promo_category FROM promotions;

   C. SELECT DISTINCT promo_category, promo_cost FROM promotions ORDER BY 1;

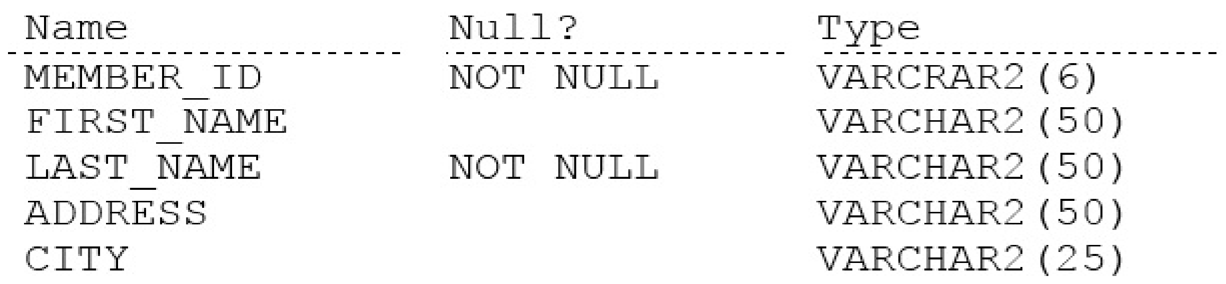
   D. SELECT promo_category DISTINCT promo_cost, FROM promotions ORDER BY 2;

   E. SELECT promo_cost, promo_category FROM promotions ORDER BY 1;

2. Which of the following statements are true about SELECT statements. Choose all that apply.

   A. Select statements are case sensitive.
   B. Select statements can be written in multiple lines
   C. Keywords cannot be abbreviated.
   D. No order for the placement of clauses (FROM ,WHERE , ORDER BY)
   E. Select statements can be used to read records from multiple tables.

3. Examine the description of the MEMBERS table:



```
 Name                    Null?              Type
----------------------- ------------------ -----------------------
 MEMBER_ID               NOT NULL           VARCRAR2(6)
 FIRST_NAME                                 VARCHAR2(50)
 LAST_NAME               NOT NULL           VARCHAR2(50)
 ADDRESS                                    VARCHAR2(50)
 CITY                                       VARCHAR2(25)
```

Examine the partial query:
SELECT city, last_name AS lname FROM members;
You want to display all cities that contain the string AN. The cities must be returned in ascending order, with the last names further sorted in descending order.
Which two clauses must you add to the query?

A. ORDER BY 1, 2

B. ORDER BY 1, lname DESC

C. WHERE city LIKE '%AN%' ORDER BY last_name DESC , city ASC

D. WHERE city = '%AN%' ORDER BY last_name , city ASC

E. WHERE city LIKE '%A%N%' ORDER BY last_name DESC , city ASC

F. WHERE city IN ('%AN%') ORDER BY last_name DESC, city ASC

4. Which two are true about the precedence of operators and conditions? (Choose two.)

A. * has a higher order of precedence than + (addition).

B. ( ) has a lesser order of precedence than any other operator.

C. NOT has a higher order of precedence than AND and OR in a condition.

D. AND and OR have the same order of precedence in a condition.

E. The order of precedence is the same as the order given in the expression.

5. Which statement is true regarding the default behavior of the ORDER BY clause

A. In a character sort values are case-sensitive.
B. NULL values are not considered at all by the sort operation.
C. Only those columns listed in the SELECT list can be used in the ORDER BY clause.
D. Numeric values are displayed from the maximum to the minimum value if they have any decimal positions

6. Examine the structure of the BOOKS_TRANSACTIONS table:

Name	Type
TRANSACTION_ID	VARCHAR2 (6)
BORROWED_DATE	DATE
DUE_DATE	DATE
BOOK_ID	INT
MEMBER_ID	VARCHAR2 (6)

You want to display the member IDs, due date, and late fee as $2 for all transactions. Which SQL statement must you execute?

A. SELECT member_id AS MEMBER_ID, due_date AS DUE_DATE, $2 AS LATE_FEE FROM BOOKS_TRANSACTIONS;
B. SELECT member_id 'MEMBER ID', due_date 'DUE DATE', '$2 AS LATE FEE' FROM BOOKS_TRANSACTIONS;

C. SELECT member_id AS "MEMBER ID", due_date AS "DUE DATE", '$2' AS "LATE FEE" FROM BOOKS_TRANSACTIONS;

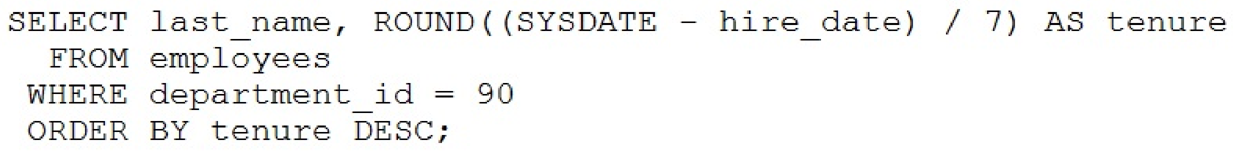
D. SELECT member_id AS "MEMBER ID", due_date AS "DUE DATE", $2 AS "LATE FEE" FROM BOOKS_TRANSACTIONS;

7. Examine the description of the EMPLOYEES table:

NAME	TYPE
_____	_____
EMPNO	INT
LAST_NAME	VARCHAR(50)
HIREDATE	DATE
SALARY	FLOATt(8,2)
DEPTNO	INT

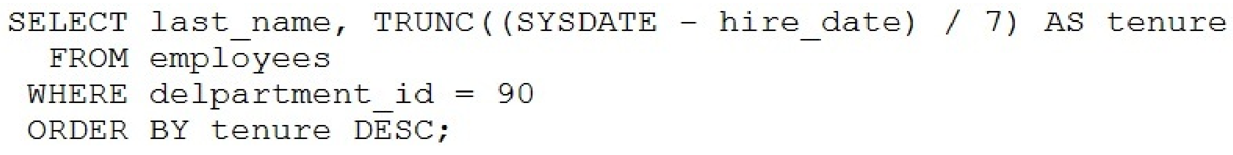
For each employee in department 90 you want to display:
1. their last name
2. the number of complete weeks they have been employed
The output must be sorted by the number of weeks, starting with the longest serving employee first.
Which statement will accomplish this?

A.



```
SELECT last_name, ROUND((SYSDATE - hire_date) / 7) AS tenure
  FROM employees
 WHERE department_id = 90
 ORDER BY tenure DESC;
```

B.



```
SELECT last_name, TRUNC((SYSDATE - hire_date) / 7) AS tenure
  FROM employees
 WHERE delpartment_id = 90
 ORDER BY tenure DESC;
```

C.

```
SELECT last_name, ROUND((SYSDATE - hire_date) / 7) AS tenure
  FROM employees
 WHERE department_id = 90
 ORDER BY tenure;
```

D.

```
SELECT last_name, TRUNC((SYSDATE - hire_date) / 7) AS tenure
  FROM employees
 WHERE department_id = 90
 ORDER BY tenure;
```

8.Examine the description of the PRODUCT_DETAILS table:

```
Name                    Null?            Type
----------------------- ---------------- ------------------
PRODUCT_ID              NOT NULL         NUMBER(2)
PRODUCT_NAME            NOT NULL         VARCHAR2(25)
PRODUCT_PRICE                            NUMBER(8,2)
EXPIRY_DATE                              DATE
```

Which two statements are true? (Choose two.)

A. EXPIRY_DATE contains the SYSDATE by default if no date is assigned to it.

B. PRODUCT_PRICE can be used in an arithmetic expression even if it has no value stored in it.

C. PRODUCT_NAME cannot contain duplicate values.

D. EXPIRY_DATE cannot be used in arithmetic expressions.

E. PRODUCT_PRICE contains the value zero by default if no value is assigned to it.

F. PRODUCT_ID can be assigned the PRIMARY KEY constraint.

9. Examine the description of the EMPLOYEES table

NAME	TYPE
EMPID	INT(3)
FIRST_NAME	VARCHAR(200)
LAST_NAME	VARCHAR(200)
SALARY	FLOAT(6,2)
DEPTNO	INT(2)

Which two queries will result in an error? (Choose two.)

A.

```
SELECT first_name last_name
  FROM employees;
```

B.

```
SELECT first_name, last name
  FROM employees;
```

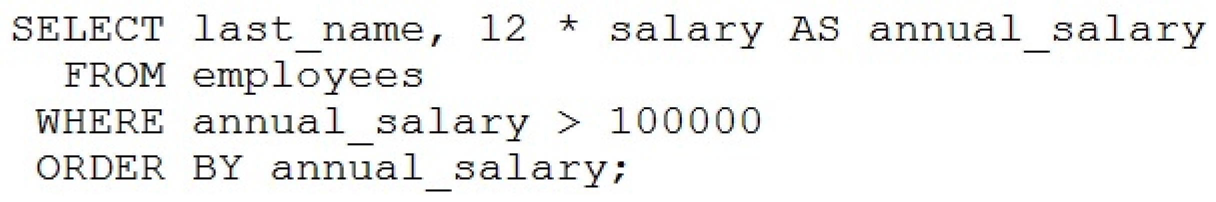
C.

```
SELECT last_name, 12 * salary AS annual_salary
  FROM employees
 WHERE annual_salary > 100000
 ORDER BY 12 * salary;
```

D.

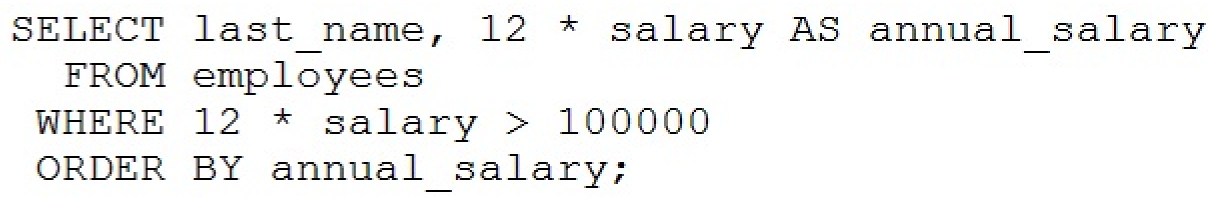
```
SELECT last_name, 12 * salary AS annual_salary
  FROM employees
 WHERE 12 * salary > 100000
 ORDER BY 12 * salary;
```

E.



```
SELECT last_name, 12 * salary AS annual_salary
   FROM employees
  WHERE annual_salary > 100000
  ORDER BY annual_salary;
```

F.



```
SELECT last_name, 12 * salary AS annual_salary
   FROM employees
  WHERE 12 * salary > 100000
  ORDER BY annual_salary;
```

10. Which statement will execute successfully, returning distinct employees with non-null first names?

A. SELECT first_name, DISTINCT last_name FROM employees WHERE first_name <> NULL;

B. SELECT first_name, DISTINCT last_name FROM employees WHERE first_name IS NOT NULL;

C. SELECT DISTINCT * FROM employees WHERE first_name IS NOT NULL;

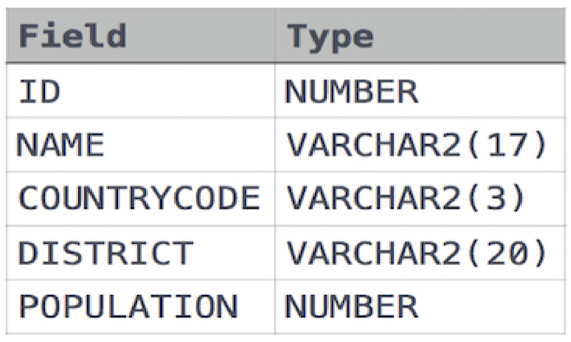
D. SELECT DISTINCT * FROM employees WHERE first_name <> NULL;

Exercises

Read the questions below and write suitable queries

1. The city table is described as follows

CITY



Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

a. Find the name , district and Population of all cities.
b. Write a query to find the district and population of all American cities with a population more than 100000. The country code for America is USA.
c. Query all the columns for a city in the city table with ID 1661.
d. Write a query to find the details of all Japanese cities . The country code for Japan is JPN.
e. Write a query to find the name and population of Brooklyn district in New York city.

2. The customer table has the following structure and the values are given below.

customer_id	cust_name	city	grade	salesman_id
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	5007
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

a. Write a query to find the names of the customers who have ‘o’ as second from the last character. Ex: Marg**o**n.
b. Find the Customer_id, cust_name,city ,grade and salesman_id of the customers who do not live in New York and have a grade value that exceeds 100.
c. Find the customers who live in California and London and have a rating from 200 to 300.
d. Find the customers who have not received any grading
e. Find the details of the customers who were attended by salesman 5002.
f. Display the names of the cities in alphabetical order and then the names of the customers in reverse order within each city.

g. Display the unique cities from the customer table.
h. Display the details of the customers who live in London and attended by salesmen either 5002 or 5005.

Answers to MCQs

1. A, C
2. B,C,E
3. C
4. A,C
5. A
6. C
7. C
8. B,F
9. C, E
10. C