

2012_April

A1

- a. parameter
- b. Cint
- c. Function
- d. pound
- e. different
- f. parentheses
- g. concatenation
- h. null
- i. right
- j data

A2:

```
SELECT customer_number, first_name, av_credit  
FROM CUSTOMER  
WHERE av_credit BETWEEN 500 AND 1000;
```

A3:

```
SELECT DISTINCT title  
FROM Author
```

A4:

- a. 9.5
- b. 27
- c. 2
- d. Computer
- e. 3

A5. CREATE

DROP

TRUNCATE

ALTER

RENAME

A6:

MAX(), MIN(), SUM(), AVG(), COUNT()

A7:

SELECT ITEM, grade * weight AS [PRICE]

FROM Stock;

B1:

- a. i. SELECT ProductId, BrandName
FROM Product
Where Quantity > 10;
- ii. SELECT BrandName, Quantity
FROM Product

ORDER BY Quantity DESC;

iii.

SELECT BrandName AS [Brands start with S]

FROM Product

Where BrandName LIKE 'S%';

- b. i. Aggregate function: It returns a single value calculated from values in a column.
- ii. Scalar function: It returns a single value based on input value.

c. LCASE(),UCASE(),ROUND(),LEN()

d. SELECT Count(*)
FROM EMPLOYEE
GROUP BY DEPTNO;

B2.

a. SELECT ISNULL(comm,0)
FROM emp;

b. i. LCASE()
ii. UCASE()
iii.INITCAP()

c.

a. NULL
b. SELECT ISNULL(INTEREST, 'Not applicable')
FROM CUSTOMER;

d. ADVANTAGES:

- i. Minimum data redundancy.
- ii. Data security and backup
- iii. Transaction processing
- iv. Data independency
- v. Multiple view

e. Employee with salary low and high will be displayed from the employee table.

f. A Cartesian product is a combination of every row of one table to every row of another table.

B3.

a.

i. SELECT*
FROM employee;

ii. SELECT empname,joindate,DateAdd(day,60,joindate) AS probation
FROM employee;

iii. SELECT COUNT(*)
FROM employee
WHERE joindate BETWEEN '2006-01-01' AND '2006-01-07';

b. i. SELECT Branch,SUM(Mem_fees)
FROM Member
GROUP BY Branch;

ii. SELECT ID,Membername
FROM MEMBER
WHERE Branch=(SELECT BRANCH
FROM MEMBER
WHERE Branch='Jason');