Question Paper Discussion

1. What are the basic data types available for attributes in SQL? (3)

Ans. The basic data types available for attributes include

- char(n): Fixed length character string, with user-specified length n.
- varchar(n): Variable length character strings, with user-specified maximum length n.
- int: Integer (a finite subset of the integers that is machine-dependent).
- smallint: Small integer (a machine-dependent subset of the integer domain type).
- numeric(p,d): Fixed point number, with user-specified precision of p digits, with d digits to the right of decimal point. (ex., numeric(3,1), allows 44.5 to be stores exactly, but not 444.5 or 0.32)
- real, double precision: Floating point and double-precision floating point numbers, with machine-dependent precision.
- float(n): Floating point number, with user-specified precision of at least n digits.

2. List the aggregate functions in SQL. (3)

Ans. Aggregate functions are used to summarize information from multiple tuples into a single-tuple summary.
Grouping is used to create sub-groups of tuples before summarization.
A number of built-in aggregate functions exist:

- COUNT function returns the number of tuples or values as specified in a query.
- SUM function returns the sum of a multiset
- MAX function returns maximum of a multiset
- MIN function returns minumum of a multiset and
- AVG function returns average of a multiset

These functions can be used in the SELECT clause or in a HAVING clause

3. Let E = {B →A, D→A, AB→D} is a set of Functional Dependencies. Find a minimal cover for E.

Ans. Step 1

All above dependencies are in canonical form that is, they have only one attribute on the right-hand side

Step 2

we need to determine if AB → D has any redundant attribute on the left-hand side;

that is, can it be replaced by B → D or A → D?

Since B → A, by augmenting with B on both sides (IR2), we have BB → AB, or B → AB (i).

However, AB → D as given (ii).

Hence by the transitive rule (IR3), we get from (i) and (ii), B→ D. Thus AB → D may be replaced by B → D.

We now have a set equivalent to original E, say

E’: {B → A, D → A, B → D}.

No further reduction is possible in step 2 since all FDs have a single attribute on the left-hand side.

Step 3

we look for a redundant FD in E’.

By using the transitive rule on B → D and D → A, we derive B→ A. Hence B → A is redundant in E’ and can be eliminated.

Therefore, the minimal cover of E is {B → D, D → A}.

4. Define Boyce-Codd normal form(BCNF). Give an example of a relation that is in 3NF but not in BCNF. (3)

Ans. Boyce-Codd normal form (BCNF) was proposed as a simpler form of 3NF, but it was found to be stricter than 3NF. That is, every relation in BCNF is also in 3NF; however, a relation in 3NF is not necessarily in BCNF. BCNF is stricter than 3NF.

A table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table.

Relation Std_tech(Student, course, teacher)

With FD={Student, course}--> teacher

Teacher-->course

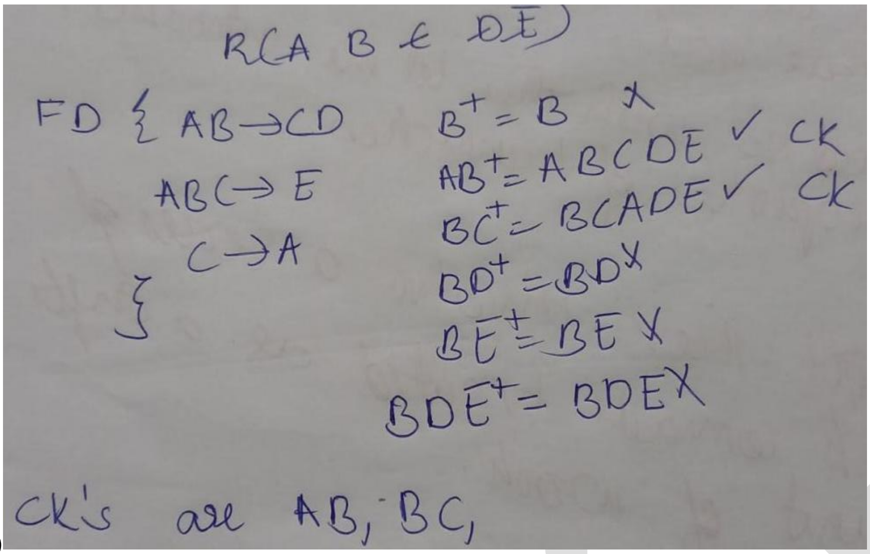
This is in 3nf not in bcnf

5. Consider the following relations for bank database (Primary keys are underlined):
Customer (customer-name, customer-street, customer-city)
Branch (branch-name, branch-city, assets)
Account (account-number, branch-name, balance)
Depositor (customer-name, account-number)
Loan (loan-number, branch-name, amount)
Answer the following in SQL:
i) Create tables with primary keys and foreign keys (5)
ii) Create an assertion for the sum of all loan amounts for each branch must be less than the sum of all account balances at the branch. (4)

Ans ii)
```
create assertion    sum-constraint check
           (not exists (select * from branch
                  where (select sum(amount) from loan
                         where (loan.bname =    branch.bname >=
                                (select sum)amount) from account
                                where (account.bname =    branch.bname)))
```

6. Given R(A,B,C,D,E) with the set of FDs, F = {AB→CD, ABC →E, C →A}.
i) Find any two candidate keys of R (3)

ii) What is the normal form of R? Justify your answer. (6)

Ans. I) 

Ii) AB→CD //SK(AB) determines prime(C)or non prime(D) allowed in 3NF
ABC →E // SK(AB) determines non prime(E) allowed in 3NF
C →A //Prime(C) determine Prime(C) allowed in 3NF
So this is in 3NF

7. a) What are Armstrong's axioms? (3)
b) Write an algorithm to compute the attribute closure of a set of attributes (X) under a set of functional dependencies (F).(3)
c) Explain three uses of attribute closure algorithm. (3)
Ans. Refer notes

8. What are the different types of single-level ordered indices? Explain. (10)
Ans. Explain Primary, Clustered and Secondary level indexing with example and figure

9. a) What is a B+-tree? (2)
b) Describe the structure of both internal and leaf nodes of a B+-tree of order p
Ans. Refer notes

10. Differentiate between static hashing and dynamic hashing. (10)
Ans. Refer notes

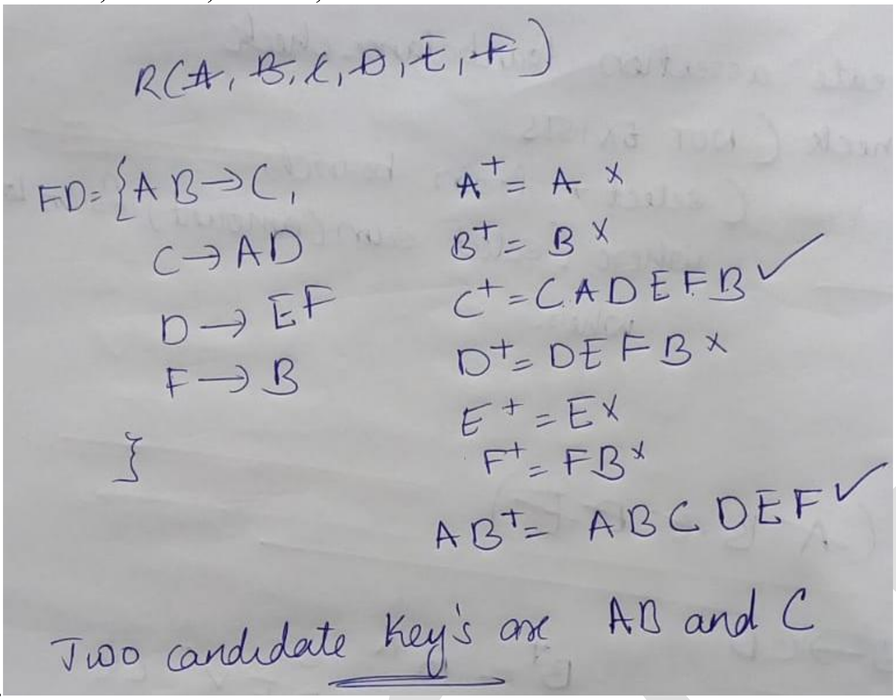
11. Illustrate the GROUP BY clause with the help of a real example. (3)

One/two sentence explanation of GROUPBY	**1 mark**
A sample table, Correct SQL expression with GROUP BY and output	**2 marks**

Note that the attributes appearing in GROUP BY clause should appear in SELECT clause also.

12. Determine any two candidate keys of the relation R(A,B,C,D,E,F) with FDs AB→C,C→AD, D→EF, F→B.

Ans.



R(A, B, C, D, E, F)

FD= {AB→C,
C→AD
D→EF
F→B
}

$A^+ = A$ ✗
$B^+ = B$ ✗
$C^+ = CADEFB$ ✓
$D^+ = DEFB$ ✗
$E^+ = E$ ✗
$F^+ = FB$ ✗
$AB^+ = ABCDEF$ ✓

Two candidate Key's are AB and C

13. Give an example for a relation that has insertion, deletion and update anomalies. Which type(s) of functional dependency can formally model these anomalies? Quote one such 0dependency from your example(3)

Any relation which is a result of combining two real-world entity sets will have these anomalies.

The student us expected to

- Give one such relation. For example, STUDENT-COURSE(<u>ROLLNO, COURSEID</u>, NAME, CLASS, CNAME, CCREDIT, GRADE). **1 mark**
- Two types of the dependencies that model it – transitive dependency and partial functional dependency. **1mark**
- Quote any one of PFD or TFD from the given relation. (Example: {ROLLNO, COURSEID } → CNAME while COURSEID → CNAME is PFD) **1 mark**

14. Illustrate the use of assertions with a typical example. (3)

Ans.

A typical simple assertion with brief explanation is expected.

Assertions are used to specify general restrictions on data stored in tables. These restrictions cannot be expressed using integrity constraints.
Example:

```
CREAT ASSERTION SALARY_CONSTRAINT
CHECK (NOT EXISTS (SELECT *
           FROM EMPLOYEE E, EMPLOYEE M,
                DEPARTMENT D
           WHERE E.SALARY > M.SALARY AND
                E.DNO=D.NUMBER AND
                D.MGRSSN=M.SSN))
```

The above assertion makes sure that salary of an employee does not exceed that of his/her manager. The assertion part comes within the *check* clause. For every update on salary of the employee, the database checks the condition given by the assertion and alarms if it fails.

example: 1.5 marks, explanation: 1.5 marks

15. Consider a relation (A,B,C,D,E,F) with A as the only key. Assume that the dependencies E→F and C→DEH hold on R.
    (i) Is R is in 2NF? If not, decompose to 2NF.
    (ii) Is R is in 3NF? If not, decompose to 3NF. (6)

Ans.

(Here the attribute H is missing. However that does not affect the approach to answer the question. The presence of H is quite irrelevant as it does not appear on the left side of any FD. C→DEH and be taken as C→DE.)

(i) There is no partial functional dependency as the only key is *not* composite. Hence the relation in 2NF. **2marks**

(ii) There are two transitive dependencies – through E→F and through C→DE. Therefore decompose the original relation into,
1. R1(A,B,C,E)
2. R2(C,D,E)
3. R3(E,F)

Primary keys underlined; foreign keys double-underlined.

Complete decomposition and a description similar to the above: **4 marks**

16. In the following tables ADVISOR and TAUGHTBYare foreign keyd referring to the table PROFESSOR. ROLLNO and COURSEID in ENROLLMENT refer to tables with primary keys of the same name.
    STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
    COURSE(COURSEID, CNAME, TAUGHTBY, CREDITS)
    PROFESSOR(PROFID,PNAME, PHONE)
    ENROLLMENT(ROLLNO, COURSEID, GRADE)
    Write SQL expressions for the following queries:
    (i) Names of courses taught by 'Prof. Raju'.
    (ii) Names of students who have not enrolled for any course taught by 'Prof. Ganapathy'.

(iii) For each course, name of the course and number of students enrolled for the course.

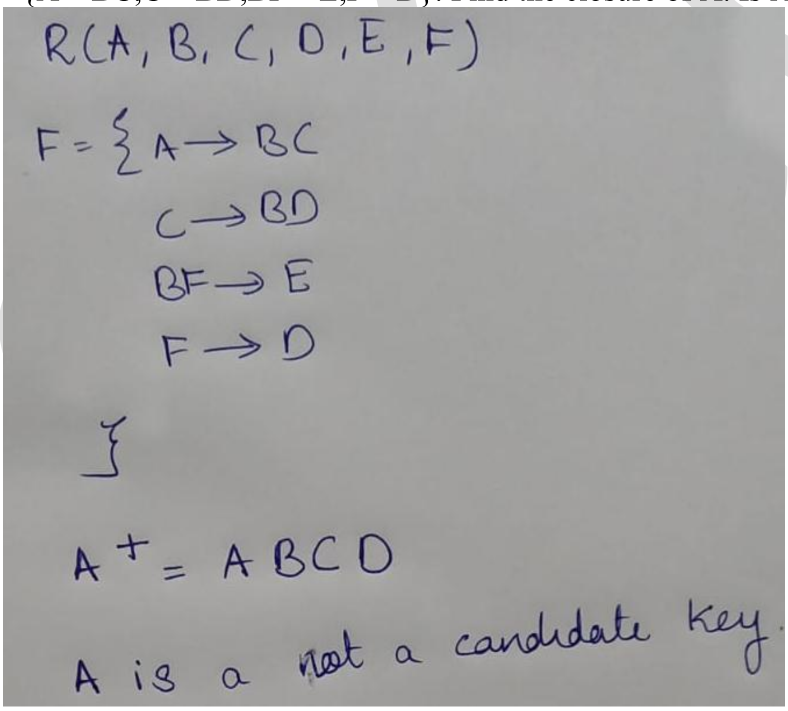
(i) SELECT C.CNAME FROM PROFESSOR P, COURSE C WHERE P.PROFID = C.TAUGHTBY AND P.PNAME = 'Prof. Raju' **2 marks**

(ii) Being a negation query, the best way to express is through nested query.
SELECT S.NAME FROM STUDENT S WHERE S.ROLLNO NOT IN (SELECT E.ROLLNO FROM ENROLLMENT E, COURSE C, PROFESSOR P WHERE E.COURSEID=C.COURSEID AND C.TAUGHTBY = P.PROFID AND P.PNAME = 'Prof. Ganapathy')
Instead of 'NOT IN', 'NOT = ANY' can also be used. **4 marks**

(iii) SELECT C.CNAME COUNT (*) FROM COURSE C, ENROLLMENT E WHERE C.COURSEID=E.COURSEID GROUP BY CNAME
or
SELECT CNAME COUNT (*) FROM COURSE NATURAL JOIN ENROLLMENT GROUP BY CNAME
or
SELECT C.CNAME COUNT (*) FROM COURSE C JOIN ENROLLMENT E ON C.COURSEID=E.COURSEID GROUP BY CNAME

3 marks

17. Consider a relation R={A,B,C,D,E,F} and a set of functional dependencies F={A→BC,C→BD,BF→E,F→D}. Find the closure of A. Is A a candidate key? Justify.(3)

Ans. 
R(A, B, C, D, E, F)

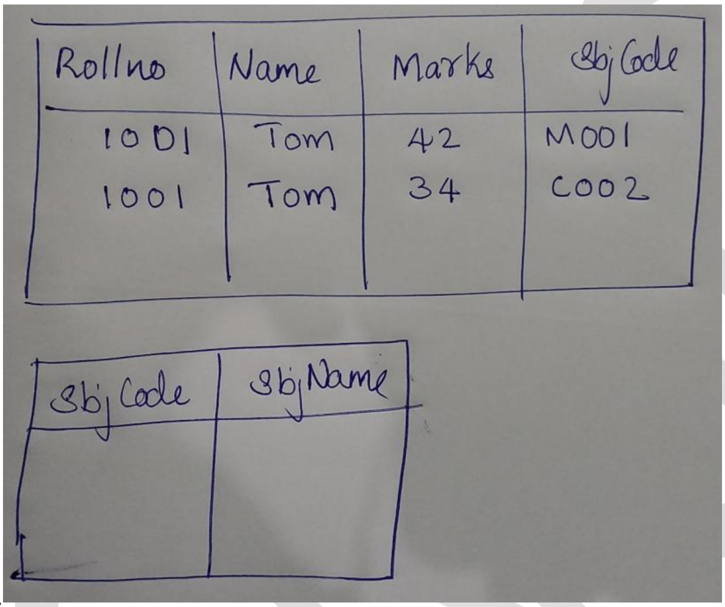
F = { A → BC
C → BD
BF → E
F → D
}

$A^+ = ABCD$

A is a not a candidate key.

18. a) Consider the following table MARKS. Why is the table not in 1NF? Reconstruct the table so that it is in 1NF. (5)

Roll No.	Name	Marks	Subject	
			Code	Name
1001	Tom	42	M001	Maths
		34	C002	Chemistry
		37	P003	Physics
1057	Sam	21	M001	Maths
		25	C002	Chemistry
		34	P003	Physics
1001	Tom	45	M001	Maths
		48	C002	Chemistry
		44	P003	Physics

Ans. 

Rollno	Name	Marks	Sbj Code
1001	Tom	42	M001
1001	Tom	34	C002

Sbj Code	Sbj Name

19. Given a relation R(A,B,C). Find the minimal cover of the set of functional dependencies given;

F= {A→BC, B→C, A→B, AB→C}

Ans.

20. Consider the relation R = {A, B, C, D, E, F, G, H} and the set of functional dependencies F = {A→DE, B→F, AB→C, C→GH, G→H}. What is the key for R? Decompose R into 2NF and then 3NF relations. (9)

Ans.

R(A, B, C, D, E, F, G, H}

FD= {A → DE , B → F , AB → C , C → GH, G → H}

First Find CK's

AB^+ = ABDEFCGH ✓ CK

Now look at FD here

A → DE
B → F } these are partial dependencies

So R not in 2NF

So lets decompose R to make it 2NF

A^+ = ADE
R_1(ADE)
FD_1 = {A → DE}
3NF

B^+ = BF
R_2(B F)
FD_2 = {B → F}
3NF

R_3(ABCGH)
FD_3 = {AB → CGH,
C → GH
G → H
}

here CK is AB
So C → GH
G → H
is not allowed in 3NF
since it is of form
non prime → nonprime
(transitive dependency)

So we need to decompose R_3

R_{31}(C, G)
in 3NF

R_{32}(G, H)
in 3NF

R_{33}(ABC)
in 3NF

So the final decomposed relation in 3NF are

R_1(A, D, E)
R_2(B, F)
R_{31}(C, G)
R_{32}(G, H)
R_{33}(A, B, C)

21. Consider the following set F of functional dependencies for relation schema R = (A, B, C, D, E).

F= {A → BC ,CD → E, B → D, E → A}

Compute the canonical cover of F. (3)

Ans. The given set of FDs F is:- A → BC CD → E B → D E → A The left side of each FD in F is unique. Also none of the attributes in the left side or right side of any of the FDs is extraneous. Therefore the canonical cover Fc is equal to F.

22. Suppose that we have an ordered file with 400,000 records stored on a disk with block size 4,096 bytes. File records are of fixed size and are unspanned,with record length 200 bytes. How many blocks are needed for the file? Approximately, how many block accesses are required for a binary search in this file? On an average, how many block accesses are required for a linear search, if the file is nonordered? (6)

Ans.

ordered file

No of records = 400 000

Block size = 4096 bytes

Unspanned

Record size = 200 bytes

No of Records per blocks (bfr) = $\frac{\text{Block size}}{\text{Record size}}$

$= \left\lfloor \frac{4096}{200} \right\rfloor = 20$

No of blocks needed = $\frac{\text{No of records}}{\text{bfr}}$

$= \left\lceil \frac{400000}{20} \right\rceil$

$= 20000$

No of block access needed for a binary search on this file

$= \log_2 b$

$= \log_2 20000$

$= \lceil 14.2 \rceil = 15$

No of block needed for linear search if file is non ordered $= \frac{b}{2} = \frac{20000}{2} = 10000$

23. Given below are two sets of FDs for a relation R(A,B,C,D,E).Are they equivalent?
F1 = {A→B, AB→C, D→AC, D→E}
F2 = {A→BC, D→AE} (5)

R(A B C D E)
F_1 = {A → B, AB → C, D → AC, D → E}
F_2 = {A → BC, D → AE}

Step 1: Checking all FDs of FD_1 are present in FD_2

✓ A → B is in FD_1 but not directly in FD_2
A → BC in F_2
but by Decomposing A → BC we can still hold this to A → B and A → C

✓ AB → C not directly in FD_2
take AB^+ in FD_2 = ABC
So this hold

✓ D → AC not directly in FD_2
D^+ = DAEBC
so this hold.

✓ D → E holds

Step 2: Check of FDs in F_2 present in F_1

✓ A → BC take A^+ = ABC ✓ hold in F1
✓ D → AE take D^+ = DACE ✓ hold in F1

So F_1 is cover of F_2 and F_2 is cover of F_1
So we can say F_1 and F_2 are equivalent

Ans.

24. Suppose that we decompose the schema R = (A, B, C, D, E) into
R1(A, B, C)
R2(A, D, E)
Test whether the given decomposition is a lossless-join decomposition, if the following set F of functional dependencies holds in R:
F= {A →BC, D →E, B → D, E → A} (5)

Ans.

R(A, B, C, D, E)

R_1(A B C) R_2(A D E)

FD = {A → B C, D → E, B → D, E → A}

	S				
	A	B	C	D	E
R_1	~~b_{11}~~ a_1	~~b_{12}~~ a_2	~~b_{13}~~ a_3	~~b_{14}~~ a_4	b_{15}
R_2	~~b_{21}~~ a_1	~~b_{22}~~ a_2	~~b_{23}~~ a_3	~~b_{24}~~ a_4	~~b_{25}~~ a_5

Now consider each FD

✓ A → BC
here col of A have same value so make BC also same. by converting b value to a

D → E
here col D has different value no action taken

B → D
here B col has same value chang D column.

E → A
here E col is different so no action taken

Here R_2 has only a values So we can conclude this is Lossless Join

25. Given a relation R(A,B,C,D,E,F,G, H) with keys BD and C and functional dependencies D→G, E→F and H→C, decompose the R into the highest normal form possible.(9)

Ans.

R(A B C D E F G H)

Key given BD, C

FD = { D → G, E → F, H → C }

here D → G is partial dependency so violate 2NF

decompose to make it 2NF

R_1(D, G) is BCNF only one FD = { D → G }

R_2(A B C D E F H)

In R_2 FD's are BD → ACEFH } these are key
C → ABDEFH
E → F
H → C

In R_2 CK's are BD and C itself.

Now look at FDs here

E → F violate 3NF
(E nonprime, F nonprime)

So decompose R_2

R_{21}(E, F) with FD_{21} = {E → F} this is in BCNF

R_{22}(A B C D E H)

with FD_{22} = { BD → ACEH,
C → ABDEH
H → C // not allowed in BCNF
(H nonprime, C prime)
}

decompose R_{22},

R_{221}(H, C) in BCNF

R_{222}(A B D E H) in BCNF

FD_{222}{BD → AEH}

So R decomposed tables are R_1(D, G) R_{21}(E, F) R_{221}(H, C) R_{222}(A B D E H)

26. Consider the following relations for bank database (Primary keys are underlined):
Customer (customer-name, customer-street, customer-city)
Branch (branch-name, branch-city, assets)
Account (account-number, branch-name, balance)
Depositor (customer-name, account-number)
Loan (loan-number, branch-name, amount)
Answer the following in SQL:
i) Create tables with primary keys and foreign keys
ii) Create an assertion for the sum of all loan amounts for each branch must be less than the sum of all account balances at the branch.

Ans.