**K226007**

**Yousha Mehdi**

**Database Assignment 1**

**BCS-5L**

Q1:

A) 1. Book

2. Book\_Authors

3. Publisher

4. Book\_copies

5. Book\_Loans

6. Libarary\_branch

7. Borrower

B) Book (Bookid)

Book\_Authors (Bookid + AuthorName)CPK

Publisher (Name + address)CPK

Book\_Copies(Book\_id+Branchid)

Book\_loans(Bookid+Branchid+CardNo)

Library\_Branch(BranchId)

Borrower(CardNo)

C) **Book\_Authors:** Foreign key Book(Bookid) To relate books with authors.

**Book:** Foreign key Publisher(Name) to ensure book publisher name.

**Book\_Copies**: Foreign keys Book(Bookid) and Library\_branch(branchid), to link number copies with the library branch.

**Book\_loans:** Foreign keys Book (Bookid), Library\_branch(branchid) and Borrower(CardNo), to record which borrower borrows which book and the branch from which it is borrowed.

D) **BOOK**:

* BookId: Integer (Unique)
* Title: String
* PublisherName: String

**BOOK\_AUTHORS**:

* BookId: Integer (Book identifier)
* AuthorName: String

**PUBLISHER**:

* Name: String (Unique)
* Address: String
* Phone: String

**BOOK\_COPIES**:

* BookId: Integer (Book identifier)
* BranchId: Integer
* No\_Of\_Copies: Integer

**BOOK\_LOANS**:

* BookId: Integer (Book identifier)
* BranchId: Integer (Library branch identifier)
* CardNo: Integer (Borrower's card number)
* DateOut: Date
* DueDate: Date

**LIBRARY\_BRANCH**:

* BranchId: Integer (Unique)
* BranchName: String
* Address: String

**BORROWER**:

* CardNo: Integer (Unique)
* Name: String
* Address: String
* Phone: String

E) Book and Book\_Author have many-to-one relationship, multiple books can have single authors.

Different book\_id’s entries will be assigned to a single author name.

Q2.

1. Select d.name

from Doctors d, operating\_room o

Where (d.d\_id = o.d\_id)

And room = ‘R-11’;

b) Select p.names, p.age

From patients p

Join Hospital h on (p.h\_id = h.h\_id)

Where p.age > 15 AND h.h\_name = ‘AKU’ AND h.location = ‘Karachi’;

c) Select Count(p.names) as Total\_patients

From patients p

Join Hospital h on (p.h\_id = h.h\_id)

Where h.h\_name = ‘AKU’;

d) Select d.name

From Doctors d

Join Operating\_Room r on (d.d\_id = o.d\_id)

Group by d.d\_name

Having count(o.room) > 1;

e) Select h.name, count(p.p\_name) As No\_of\_patients

From Hospital h, Patients p

Where (p.h\_id = h.h\_id)

Group by h.h\_name

Order by No\_of\_Patients desc;

f) Select p.names

From patients p

Join Hospital h on (p.h\_id = h.h\_id)

Where h.h\_name <> ‘AKU’;

g) Select p.names

From patients p

Join Hospital h on (p.h\_id = h.h\_id)

Where h.h\_name = ‘Liauat’ And h.location = ‘Lahore’;

h) select p.p\_num, p.p\_name

From patient p

Join Surgery\_detail s on (s.p\_num = p.p\_num)

Join Operation\_Room o on (s.o\_name = o.o\_name)

Join Doctor d on ( o.d\_id = d.d\_id )

Where d.name = ‘Dr. Muhammad Rafi’;

i) Select d.name

from Doctors d

Join operating\_room o on (d.d\_id = o.d\_id)

Where room <> ‘R-109’;

J) Select p.p\_name

From Patient p

Join Surgery\_datails s on (p.p\_num = s.p\_num)

Group by p.p\_name

Order by Count(s.o\_name) Desc;

Q3.

A) Select b.title

From Books b

Join Reviews r on (b.book\_id = r.book\_id)

Where r.rating = 5;

B) Select a.author\_name

From Authors a

Join Books b on (a.author\_id = b.author\_id)

Where b.genre = ‘Science Fiction’

Group by a.author\_name

Having count(b.book\_id) > 3;

C) Select c.customer\_name

From customers c

Join Orders o on (c.customer\_id = o.customer\_id)

Where o.total\_amount > 500

And month(o.order\_date) = month(curdate());

D) select b.title

From books b

Join authors a on (b.author\_id = a.author\_id)

Where b.price < (select avg(price) from books);

E) select b.title

From books b, OrderDetails o

Where b.book\_id = o.book\_id(+)

And o.book\_id = NULL;

F) select c.customer\_name

From customers c

Join reviews r on(c.customer\_id = r.customer\_id)

Group by r.review\_id having (count(r.review\_id) >1

And r.rating < 3);

G) SELECT a.author\_name, COUNT(b.book\_id) AS Total\_Books

FROM Authors a

JOIN Books b ON a.author\_id = b.author\_id

GROUP BY a.author\_name;

H) SELECT c.customer\_name

FROM Customers c

WHERE MONTH(c.join\_date) = (SELECT MONTH(c2.join\_date)

FROM Customers c2

JOIN Orders o ON c2.customer\_id = o.customer\_id

ORDER BY o.total\_amount DESC LIMIT 1)

AND YEAR(c.join\_date) = (SELECT YEAR(c2.join\_date)

FROM Customers c2

JOIN Orders o ON c2.customer\_id = o.customer\_id

ORDER BY o.total\_amount DESC LIMIT 1);

I) SELECT b.genre

FROM Books b

JOIN Inventory i ON b.book\_id = i.book\_id

WHERE i.stock\_quantity < i.reorder\_level;

J) SELECT a.author\_name

FROM Authors a

JOIN Books b ON a.author\_id = b.author\_id

LEFT JOIN Reviews r ON b.book\_id = r.book\_id

WHERE r.review\_id IS NULL;

Q4.

1)

A) Referential integrity constraint is a condition which ensures that a foreign key of one table is a primary key of another table which help the database to maintain relation between two tables.

B) Foreign key can be null if the related data is optional to have.

For example,

1. an employees table have manager\_id as a foreign key

If an employee does not have a manger assigned, then the foreign key which is manager\_id will be null.

2. student table with a foreign key of adviser\_id, if adviser is not assigned to student then adviser\_id can be null.

C) casecade new values to related child table, set null to the child table values, restrict the deletion/Modification of parent table key if child table have related rows, set default values to child table.

2)

**Inherent Model-Based Constraints**: These are fundamental to the database model itself (e.g., relational model requirement that a table must have a primary key).

**Schema-Based Constraints**: Defined at the schema level (e.g., foreign keys, data types, and domain constraints).

**Application-Based Constraints**: Enforced by the application logic rather than the database itself (e.g., business rules that are specific to the application's requirements)

1. A student table with student\_id and Email as super key, both unique, as super key is a combination of two unique keys. A primary key can be a unique key as it also have unique value it is also called minimal superkey. Minimal key can not be reduced more further.