# Lecture 2: Question Bank

**Topics:** "Data Representation, Integrity Constraints, SQL: Creating and Managing Tables, SQL: Constraints"

| | |
|---|---|
| **Q1** | Can you provide a practical use case where understanding the degree and cardinality of relations is crucial in designing a database? |
| **Q2** | What is the fundamental concept in the relational model for representing data, and how does it differ from other data models? |
| **Q3** | Explain the significance of the relation schema in the context of a relational database. Provide an example. |
| **Q4** | How does a relation instance differ from a relation schema, and how are they related? |
| **Q5** | In the context of a relational database, what is the role of a relational database? |
| **Q6** | Define the terms "degree" and "cardinality" in the context of a relation. Provide an example. |
| **Q7** | How does the relational model support data integrity, and what are the key constraints involved? |
| **Q8** | Provide an example scenario where a referential integrity constraint is crucial for data consistency. |
| **Q9** | How does a legal instance differ from an illegal instance in the context of a database? |
| **Q10** | What distinguishes a candidate key from a superkey, and why is it considered "minimal"? |
| **Q11** | Why is it essential to enforce referential integrity constraints when dealing with foreign keys? |
| **Q12** | Explain the concept of a superkey in a relational database and provide an example. |
| **Q13** | What is the role of an alternate key in a database, and how does it relate to primary keys? |
| **Q14** | How is a primary key different from a candidate key in a relation? |
| **Q15** | Define a foreign key and provide an example scenario where it is used in a relational database. |
| **Q16** | You plan to move data from a flat file to a table in your database. You decide to use SQL*Loader direct path load method to perform this task. The table in which you plan to load data in an important table having various integrity constraint defined on it. Which constraints will remain enabled by default during this operation? (Choose all that apply)<br>A. CHECK<br>B. UNIQUE<br>C. NOT NULL<br>D. PRIMARY KEY<br>E. FOREIGN KEY |
| **Q17** | The TRANS_SUMMARY table contains product-wise transaction details that get updated with every transaction in the system. Each row has cumulative transaction details of a single product and every product is identified by a product code, which is the primary key. As part of the archival process, the company wants to transfer the rows in the TRANS_SUMMARY table to the TRANS_SUMMARY_DUP table at the end of |

| | |
|---|---|
| | every quarter of the year. Along with existing products, the company deals with many new products during every quarter. Which method is best suited for this quarterly data transfer?<br>A. Using the MERGE command<br>B. Using the SQL*Loader utility<br>C. Using the correlated UPDATE command<br>D. Using the INSERT command to perform bulk operation |
| **Q18** | You execute this command to drop the ITEM table, which has the primary key referred in the ORDERS table: SQL> DROP TABLE scott.item CASCADE CONSTRAINTS PURGE; Which two statements are true about the effect of the command? (Choose two.)<br>A. No flashback is possible to bring back the ITEM table.<br>B. The ORDERS table is dropped along with the ITEM table.<br>C. The dependent referential integrity constraints in the ORDERS table are disabled.<br>D. The dependent referential integrity constraints in the ORDERS table are removed.<br>E. The table definition of the ITEM table and associated indexes are placed in the recycle bin |
| **Q19** | If you have a column that will only have values between 1 and 250 what data type will you use? |
| **Q20** | Consider a hypothetical database for a library management system. The database has a table named "Books" that stores information about various books in the library. The current structure of the "Books" table is as follows:<br>SQL<br>CREATE TABLE Books (<br>   BookID INT PRIMARY KEY,<br>   Title VARCHAR(100),<br>   Author VARCHAR(50),<br>   Genre VARCHAR(30),<br>   PublishedYear INT,<br>   Quantity INT<br>);<br>You've been tasked with making modifications to the "Books" table using the "ALTER TABLE" statement to accommodate additional information.<br><br>**Perform the following tasks:**<br>**1.** Add a new column named "ISBN" to the "Books" table. The ISBN should be a unique identifier for each book and should be a VARCHAR data type with a maximum length of 20 characters.<br><br>**2.** The "PublishedYear" column currently uses the INT data type. Change the data type of the "PublishedYear" column to SMALLINT.<br><br>**3.** Due to a change in library policies, the "Quantity" information is no longer needed. Remove the "Quantity" column from the "Books" table.<br>Provide the SQL queries for each task in the appropriate order. |
| **Q21** | Consider the following relations:<br>Student(snum: integer, sname: string, major: string, |

level: string, age: integer)
Class(name: string, meets at: time, room: string, fid: integer)
Enrolled(snum: integer, cname: string)
Faculty(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.

Express each of the following integrity constraints in SQL unless it is implied by the primary and foreign key constraint; if so, explain how it is implied. If the constraint cannot be expressed in SQL, say so.
For each constraint, state what operations (inserts, deletes, and updates on specific relations) must be monitored to enforce the constraint.

1. Every class has a minimum enrollment of 5 students and a maximum enrollment of 30 students.
2. At least one class meets in each room.
3. Every faculty member must teach at least two courses.
4. Only faculty in the department with deptid=33 teach more than three courses.
5. Every student must be enrolled in the course called Math101.
6. The room in which the earliest scheduled class (i.e., the class with the smallest meets at value) meets should not be the same as the room in which the latest scheduled class meets.
7. Two classes cannot meet in the same room at the same time.
8. The department with the most faculty members must have fewer than twice the number of faculty members in the department with the fewest faculty members.
9. No department can have more than 10 faculty members.
10. A student cannot add more than two courses at a time (i.e., in a single update).
11. The number of CS majors must be more than the number of Math majors.
12. The number of distinct courses in which CS majors are enrolled is greater than the number of distinct courses in which Math majors are enrolled.
13. The total enrollment in courses taught by faculty in the department with deptid=33 is greater than the number of Math majors.
14. There must be at least one CS major if there are any students whatsoever.

## ANSWERS

**Q1. Can you provide a practical use case where understanding the degree and cardinality of relations is crucial in designing a database?**

*Ans:* Consider a scenario where designing a customer order database. The degree would include attributes like order_id, customer_id, product_id, quantity, and total_amount. The cardinality would vary based on the number of orders placed, reflecting the number of rows in the Orders relation. Understanding these concepts is crucial for designing a scalable and efficient database structure.

**Q2. What is the fundamental concept in the relational model for representing data, and how does it differ from other data models?**

*Ans:* The fundamental concept in the relational model is a "relation." It differs from other models by organizing data into tables with predefined schemas (relation schemas) and instances (relation instances), providing a structured and efficient way to represent and manage data.

**Q3. Explain the significance of the relation schema in the context of a relational database. Provide an example.**

*Ans:* The relation schema describes the structure of a table, specifying column names and their data types. For instance, consider the relation schema for a Students table:

Students (sid: varchar, name: varchar, login: varchar, age: integer, gpa: float)

It defines the attributes (columns) such as student ID (sid), name, login, age, and GPA.

**Q4. How does a relation instance differ from a relation schema, and how are they related?**

*Ans:* A relation schema defines the structure of a table, while a relation instance is the actual table containing data. The schema serves as a blueprint, and multiple instances (tables) can conform to the same schema, each representing a distinct set of data.

**Q5. In the context of a relational database, what is the role of a relational database?**

*Ans:* A relational database is a collection of relations (tables) with distinct names. It serves as a structured storage mechanism, allowing efficient organization, retrieval, and manipulation of data through SQL queries.

**Q6. Define the terms "degree" and "cardinality" in the context of a relation. Provide an example.**

*Ans:* The degree of a relation is the number of fields or columns. For example, in a Students relation, the degree is five, corresponding to the columns sid, name, login, age, and gpa.

The cardinality of a relation instance is the number of tuples or rows. In the given instance with six rows, the cardinality is six, reflecting the number of records.

**Q7. How does the relational model support data integrity, and what are the key constraints involved?**

*Ans:* The relational model supports data integrity through constraints such as primary keys, foreign keys, and unique constraints. These constraints ensure that data relationships are maintained, preventing inconsistencies and ensuring accuracy.

**Q8. Provide an example scenario where a referential integrity constraint is crucial for data consistency.**

*Ans:* Consider a scenario where a "Customer" table has a foreign key referencing the primary key "OrderID" in an "Orders" table. Enforcing referential integrity prevents deleting an order record if there are corresponding customer records, ensuring that no order references a non-existent customer.

**Q9. How does a legal instance differ from an illegal instance in the context of a database?**

*Ans:* A legal instance is a database instance that satisfies all the integrity constraints specified on the database schema. In contrast, an illegal instance violates one or more integrity constraints, and a well-designed DBMS prevents the storage of such instances.

**Q10. What distinguishes a candidate key from a superkey, and why is it considered "minimal"?**

*Ans:* A candidate key is a minimal superkey, meaning it is a set of attributes forming a superkey, and none of its subsets is a superkey. It is minimal because it satisfies the uniqueness requirements without unnecessary attributes.

**Q11. Why is it essential to enforce referential integrity constraints when dealing with foreign keys?**

*Ans:* Enforcing referential integrity constraints ensures that relationships between tables are maintained, preventing inconsistencies when manipulating data. It prevents actions that would create orphaned records or violate the established relationships.

**Q12. Explain the concept of a superkey in a relational database and provide an example.**

*Ans:* A superkey is a set of one or more attributes whose values uniquely identify tuples in a relation. For instance, in a relation "instructor," the combination of attributes like ID and name can serve as a superkey since it uniquely identifies each instructor

**Q13. What is the role of an alternate key in a database, and how does it relate to primary keys?**

*Ans:* An alternate key is a secondary candidate key that can also identify a row uniquely. It serves as an alternative to the primary key and can be used for unique identification if needed.

**Q14. How is a primary key different from a candidate key in a relation?**

*Ans:* The primary key is a chosen candidate key, selected by the database designer as the principal means of uniquely identifying tuples within a relation. While a relation can have multiple candidate keys, only one is designated as the primary key.

**Q15. Define a foreign key and provide an example scenario where it is used in a relational database.**

*Ans:* A foreign key in a relation refers to the primary key of another relation, establishing a relationship between them. For example, a "Student" table may have a foreign key, such as "CourseID," referencing the primary key "CourseID" in a "Courses" table.

**Q16**. **You plan to move data from a flat file to a table in your database. You decide to use SQL*Loader direct path load method to perform this task. The table in which you plan to load data in an important table having various integrity constraint defined on it. Which constraints will remain enabled by default during this operation? (Choose all that apply)**

A. CHECK

B. UNIQUE

C. NOT NULL

D. PRIMARY KEY

E. FOREIGN KEY

**Ans:** B, C, D

**Q17. The TRANS_SUMMARY table contains product-wise transaction details that get updated with every transaction in the system. Each row has cumulative transaction details of a single product and every product is identified by a product code, which is the primary key. As part of the archival process, the company wants to transfer the rows in the TRANS_SUMMARY table to the TRANS_SUMMARY_DUP table at the end of every quarter of the year. Along with existing products, the company deals with many new products during every quarter. Which method is best suited for this quarterly data transfer?**

A. Using the MERGE command

B. Using the SQL*Loader utility

C. Using the correlated UPDATE command

D. Using the INSERT command to perform bulk operation

*Ans:* A

**Q18. You execute this command to drop the ITEM table, which has the primary key referred in the ORDERS table: SQL> DROP TABLE scott.item CASCADE CONSTRAINTS PURGE; Which two statements are true about the effect of the command? (Choose two.)**

A. No flashback is possible to bring back the ITEM table.

B. The ORDERS table is dropped along with the ITEM table.

C. The dependent referential integrity constraints in the ORDERS table are disabled.

D. The dependent referential integrity constraints in the ORDERS table are removed.

E. The table definition of the ITEM table and associated indexes are placed in the recycle bin

*Ans:* A, D

**Q19. If you have a column that will only have values between 1 and 250 what data type will you use?**

*Ans:* If you are using SQL Server database then you can use the TINYINT datatype which can accommodate numbers between 0 and 255 and it needs 1 byte for storage.

**Q20. Consider a hypothetical database for a library management system. The database has a table named "Books" that stores information about various books in the library. The current structure of the "Books" table is as follows:**

SQL

```
CREATE TABLE Books (
   BookID INT PRIMARY KEY,
   Title VARCHAR(100),
   Author VARCHAR(50),
   Genre VARCHAR(30),
   PublishedYear INT,
   Quantity INT
);
```

You've been tasked with making modifications to the "Books" table using the "ALTER TABLE" statement to accommodate additional information.

**Perform the following tasks:**

**1. Add a new column named "ISBN" to the "Books" table. The ISBN should be a unique identifier for each book and should be a VARCHAR data type with a maximum length of 20 characters.**

*Ans:* Add a New Column

```
ALTER TABLE Books
ADD COLUMN ISBN VARCHAR(20) UNIQUE;
```

**2. The "PublishedYear" column currently uses the INT data type. Change the data type of the "PublishedYear" column to SMALLINT.**

*Ans:* ALTER TABLE Books

ALTER COLUMN PublishedYear SMALLINT;

**3. Due to a change in library policies, the "Quantity" information is no longer needed. Remove the "Quantity" column from the "Books" table.**

*Ans:* ALTER TABLE Books

DROP COLUMN Quantity;

**Q21. Consider the following relations:**

Student (snum: integer, sname: string, major: string, level: string, age: integer)

Class (name: string, meets at: time, room: string, fid: integer)

Enrolled (snum: integer, cname: string)

Faculty (fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.

Express each of the following integrity constraints in SQL unless it is implied by the primary and foreign key constraint; if so, explain how it is implied. If the constraint cannot be expressed in SQL, say so.

For each constraint, state what operations (inserts, deletes, and updates on specific relations) must be monitored to enforce the constraint.

**1. Every class has a minimum enrollment of 5 students and a maximum enrollment of 30 students.**

*Ans:* The Enrolled table should be modified as follows:

```
CREATE TABLE Enrolled ( snum INTEGER,
cname CHAR(20),
PRIMARY KEY (snum, cname),
FOREIGN KEY (snum) REFERENCES Student),
FOREIGN KEY (cname) REFERENCES Class,),
CHECK (( SELECT COUNT (E.snum)
FROM Enrolled E
GROUP BY E.cname) >= 5),
CHECK (( SELECT COUNT (E.snum)
FROM Enrolled E
GROUP BY E.cname) <= 30))
```

**2. At least one class meets in each room.**

*Ans:* This constraint is already guaranteed because rooms are associated with classes, and thus a new room cannot be declared without an associated class in it.

**3. Every faculty member must teach at least two courses.**

*Ans:* Create an assertion as follows:

```
CREATE ASSERTION TeachTwo
CHECK ( ( SELECT COUNT (*)
FROM Facult F, Class C
WHERE F.fid = C.fid
GROUP BY C.fid
HAVING COUNT (*) < 2) = 0)
```

**4. Only faculty in the department with deptid=33 teach more than three courses.**

*Ans:* Create an assertion as follows:

```
CREATE ASSERTION NoTeachThree
CHECK ( ( SELECT COUNT (*)
FROM Facult F, Class C
WHERE F.fid = C.fid AND F.deptid = 33
GROUP BY C.fid
HAVING COUNT (*) > 3) = 0)
```

**5. Every student must be enrolled in the course called Math101.**

*Ans:* Create an assertion as follows:

```
CREATE ASSERTION InMath101
CHECK (( SELECT COUNT (*)
FROM Student S
WHERE S.snum NOT IN ( SELECT E.snum
FROM Enrolled E
WHERE E.cname = ’Math101’)) = 0)
```

**6. The room in which the earliest scheduled class (i.e., the class with the smallest meets at value) meets should not be the same as the room in which the latest scheduled class meets.**

*Ans:* The Class table should be modified as follows:

```
CREATE TABLE Class ( name CHAR(20),
meets at TIME,
room CHAR(10),
fid INTEGER,
PRIMARY KEY (name),
FOREIGN KEY (fid) REFERENCES Faculty),
CHECK ( (SELECT MIN (meets at)
FROM Class) <>
(SELECT MAX (meets at)
FROM Class)))
```

**7. Two classes cannot meet in the same room at the same time.**

*Ans:* The Class table should be modified as follows:

```
CREATE TABLE Class ( name CHAR(20),
meets at TIME,
room CHAR(10),
fid INTEGER,
PRIMARY KEY (name),
FOREIGN KEY (fid) REFERENCES Faculty),
CHECK ((SELECT COUNT (*)
FROM ( SELECT C.room, C.meets
FROM Class C
GROUP BY C.room, C.meets
HAVING COUNT (*) > 1)) = 0))
```

**8. The department with the most faculty members must have fewer than twice the number of faculty members in the department with the fewest faculty members.**

*Ans:* The Faculty table should be modified as follows:

```
CREATE TABLE Faculty ( fid INTEGER,
fname CHAR(20),
deptid INTEGER,
PRIMARY KEY (fnum),
CHECK ( (SELECT MAX (*)
FROM ( SELECT COUNT (*)
FROM Faculty F
GROUP BY F.deptid))
< 2 *
(SELECT MIN (*)
FROM ( SELECT COUNT (*)
FROM Faculty F
GROUP BY F.deptid))))
```

**9. No department can have more than 10 faculty members.**

*Ans:* The Faculty table should be modified as follows:

```
CREATE TABLE Faculty (fid INTEGER,
fname CHAR(20),
deptid INTEGER,
PRIMARY KEY (fnum),
CHECK ( ( SELECT COUNT (*)
FROM Faculty F
GROUP BY F.deptid
HAVING COUNT (*) > 10) = 0))
```

**10. A student cannot add more than two courses at a time (i.e., in a single update).**

*Ans:* This constraint cannot be done because integratey constraints and assertions only affect the content of a table, not how that content is manipulated.

**11. The number of CS majors must be more than the number of Math majors.**

*Ans:* The Student table should be modified as follows:

```
CREATE TABLE Student ( snum INTEGER,
sname CHAR(20),
major CHAR(20),
level CHAR(20),
age INTEGER,
PRIMARY KEY (snum),
CHECK ( (SELECT COUNT (*)
FROM Student S
WHERE S.major = ’CS’) >
(SELECT COUNT (*)
FROM Student S
WHERE S.major = ’Math’)))
```

**12. The number of distinct courses in which CS majors are enrolled is greater than the number of distinct courses in which Math majors are enrolled.**

*Ans:* Create an assertion as follows:

```
CREATE ASSERTION MoreCSMajors
CHECK ( (SELECT COUNT (E.cname)
FROM Enrolled E, Student S
WHERE S.snum = E.snum AND S.major = ’CS’) >
(SELECT COUNT (E.cname)
FROM Enrolled E, Student S
WHERE S.snum = E.snum AND S.major = ’Math’))
```

**13. The total enrollment in courses taught by faculty in the department with deptid=33 is greater than the number of Math majors.**

*Ans:* Create an assertion as follows:

```
CREATE ASSERTION MoreEnrolledThanMath
CHECK ( (SELECT COUNT (E.snum)
FROM Enrolled E, Faculty F, Class C
WHERE E.cname = C.name
AND C.fid = F.fid AND F.deptid = 33) >
(SELECT COUNT (E.snum)
FROM Student S
WHERE S.major = ’Math’))
```

**14. There must be at least one CS major if there are any students whatsoever.**

*Ans:* The Student table should be modified as follows:

```
CREATE TABLE Student ( snum INTEGER,
sname CHAR(20),
major CHAR(20),
```