Question 1: Which of the following defines the structure and organization of data in a database?

A) Data integrity

B) Data security

C) Data model

D) Data abstraction

Answer: C) Data model

Question 2: In an ER diagram, what does a diamond-shaped symbol represent?

A) Entity

B) Attribute

C) Relationship

D) Key

Answer: C) Relationship

Question 3: Which of the following is NOT a level of data abstraction in a DBMS?

A) Physical level

B) Logical level

C) Conceptual level

D) Application level

Answer: D) Application level

Question 4: What is the purpose of referential integrity?

A) To ensure that each attribute contains unique values

B) To enforce business rules and logic

C) To maintain relationships between related tables

D) To ensure that data is stored in the correct order

Answer: C) To maintain relationships between related tables

Question 5: Which data model represents data in a hierarchical structure with parent-child relationships?

A) Relational model

B) ER model

C) Hierarchical model

D) Network model

Answer: C) Hierarchical model

Question 6: Which type of integrity constraint ensures that a primary key value cannot be null?

A) Domain constraint

B) Referential integrity constraint

C) Entity integrity constraint

D) Key constraint

Answer: C) Entity integrity constraint

Question 7: In an ER diagram, an oval-shaped symbol represents:

A) Entity

B) Attribute

C) Relationship

D) Key

Answer: A) Entity

Question 8: Which data model represents data as a collection of objects, along with their attributes and methods?

A) Object-Oriented model

B) Relational model

C) Entity-Relationship model

D) Hierarchical model

Answer: A) Object-Oriented model

Question 9: Which type of key uniquely identifies a record within its own table?

A) Super key

B) Candidate key

C) Primary key

D) Foreign key

Answer: C) Primary key

Question 10: Which of the following is an example of an integrity constraint?

A) Ensuring referential integrity between tables

B) Setting up user access permissions

C) Defining indexes for faster data retrieval

D) Storing data in a normalized form

Answer: A) Ensuring referential integrity between tables

Question 11: In a relational database, what does a functional dependency represent?

A) A relationship between two tables

B) A relationship between a primary key and a foreign key

C) A relationship between attributes within a single table

D) A relationship between different databases

Answer: C) A relationship between attributes within a single table

Question 12: In the context of functional dependencies, if attribute B is functionally dependent on attribute A, how is this relationship denoted?

A) A → B

B) B → A

C) A ↔ B

D) A ⊆ B

Answer: A) A → B

Question 13: Which of the following is a key property of a candidate key in a relational database?

A) It can contain null values.

B) It is a super key.

C) It uniquely identifies a tuple within a table.

D) It is a foreign key.

Answer: C) It uniquely identifies a tuple within a table.

Question 14: What is the purpose of normalizing a relational database?

A) To introduce redundancy for better performance.

B) To combine multiple tables into a single table.

C) To reduce data integrity.

D) To eliminate data anomalies and improve efficiency.

Answer: D) To eliminate data anomalies and improve efficiency.

**Here are 10 multiple-choice questions along with their answers based on normalization in a database management system (DBMS):**

Question 1: What is the primary goal of database normalization?

A) To eliminate all redundancy in the database.

B) To increase data storage capacity.

C) To simplify complex queries.

D) To improve data integrity and reduce anomalies.

Answer: D) To improve data integrity and reduce anomalies.

Question 2: In which normal form is a relation if all non-prime attributes are fully functionally dependent on the primary key?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: B) Second Normal Form (2NF)

Question 3: Which normal form ensures that every non-prime attribute is non-transitively dependent on the primary key?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: C) Third Normal Form (3NF)

Question 4: Which of the following is a potential drawback of excessive normalization?

A) Improved query performance.

B) Increased storage requirements.

C) Simplified data maintenance.

D) Enhanced data integrity.

Answer: B) Increased storage requirements.

Question 5: Which normal form is based on functional dependency and multivalued dependency concepts?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Fourth Normal Form (4NF)

Answer: D) Fourth Normal Form (4NF)

Question 6: Which normal form addresses partial dependency?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: B) Second Normal Form (2NF)

Question 7: In which normal form is a relation if it is in Second Normal Form (2NF) and has no partial dependencies?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: D) Boyce-Codd Normal Form (BCNF)

Question 8: Which of the following normal forms allows for the presence of transitive dependencies?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: C) Third Normal Form (3NF)

Question 9: Which normal form eliminates transitive dependencies as well as partial dependencies?

A) Third Normal Form (3NF)

B) Boyce-Codd Normal Form (BCNF)

C) Fourth Normal Form (4NF)

D) Fifth Normal Form (5NF)

Answer: B) Boyce-Codd Normal Form (BCNF)

Question 10: What is the key principle of normalization in database design?

A) Minimizing the number of tables.

B) Maximizing data redundancy.

C) Balancing data and index storage.

D) Reducing data duplication and anomalies.

Answer: D) Reducing data duplication and anomalies.

Question 1:

Which SQL keyword is used to retrieve data from a database table?

a) FETCH

b) SELECT

c) GET

d) EXTRACT

Answer: b) SELECT

Question 2:

Which clause is used to filter rows in a SQL SELECT statement?

a) FILTER

b) WHERE

c) HAVING

d) LIMIT

Answer: b) WHERE

Question 3:

Which SQL function is used to find the total number of rows in a table?

a) COUNT()

b) SUM()

c) AVG()

d) MAX()

Answer: a) COUNT()

Question 4:

Which SQL command is used to add new rows to a table?

a) CREATE

b) INSERT

c) UPDATE

d) ALTER

Answer: b) INSERT

Question 5:

Which clause is used to sort the result set in ascending or descending order?

a) ORDER BY

b) SORT BY

c) GROUP BY

d) ARRANGE BY

Answer: a) ORDER BY

Question 6:

What is the purpose of the SQL GROUP BY clause?

a) To filter rows based on a condition

b) To join multiple tables

c) To aggregate data and perform calculations on groups of rows

d) To retrieve data from multiple tables

Answer: c) To aggregate data and perform calculations on groups of rows

Question 7:

Which SQL statement is used to modify existing data in a table?

a) UPDATE

b) ALTER

c) MODIFY

d) CHANGE

Answer: a) UPDATE

Question 8:

Which SQL clause is used to combine rows from different tables?

a) UNION

b) JOIN

c) CONNECT

d) COMBINE

Answer: b) JOIN

Question 9:

Which SQL function is used to find the highest value in a column?

a) HIGH()

b) TOP()

c) MAX()

d) HIGHEST()

Answer: c) MAX()

Question 10:

Which SQL command is used to remove a table from a database?

a) DROP

b) DELETE

c) REMOVE

d) ERASE

Answer: a) DROP

Question 11:

What is the purpose of the SQL HAVING clause?

a) To filter rows based on a condition

b) To sort the result set

c) To filter groups based on a condition

d) To limit the number of rows returned

Answer: c) To filter groups based on a condition

Question 12:

Which SQL operator is used in the WHERE clause to specify multiple possible values?

a) AND

b) OR

c) NOT

d) XOR

Answer: b) OR

Question 13:

Which SQL function is used to calculate the average value of a column?

a) AVG()

b) AVERAGE()

c) MEAN()

d) TOTAL()

Answer: a) AVG()

Question 14:

In a SQL SELECT statement, what is the purpose of the DISTINCT keyword?

a) It sorts the result set in ascending order

b) It retrieves only the first few rows from the table

c) It removes duplicate rows from the result set

d) It groups rows based on a specified column

Answer: c) It removes duplicate rows from the result set

Question 15:

Which SQL command is used to delete rows from a table?

a) ERASE

b) REMOVE

c) DELETE

d) DROP

Answer: c) DELETE

Question 16:

Which SQL clause is used to limit the number of rows returned by a query?

a) LIMIT

b) TOP

c) ROWS

d) COUNT

Answer: a) LIMIT

Question 17:

What is the purpose of the SQL ORDER BY clause?

a) To filter rows based on a condition

b) To join multiple tables

c) To group rows based on a column

d) To sort rows in the result set

Answer: d) To sort rows in the result set

Question 18:

Which SQL function is used to find the lowest value in a column?

a) MIN()

b) SMALL()

c) LEAST()

d) BOTTOM()

Answer: a) MIN()

Question 19:

Which SQL command is used to create a new database?

a) NEW DATABASE

b) CREATE DATABASE

c) ADD DATABASE

d) MAKE DATABASE

Answer: b) CREATE DATABASE

Question 20:

What is the purpose of the SQL AS keyword?

a) To add a new column to a table

b) To rename a column or table

c) To create a new table

d) To delete rows from a table

Answer: b) To rename a column or table

## MCQ ( PART-2)

Question 1:

Which SQL join type returns all rows from both tables, filling in NULL values for non-matching rows?

a) INNER JOIN

b) LEFT JOIN

c) RIGHT JOIN

d) FULL OUTER JOIN

Answer: d) FULL OUTER JOIN

Question 2:

Which SQL clause is used to define conditions for a specific group of rows?

a) WHERE

b) HAVING

c) GROUP WHERE

d) FILTER

Answer: b) HAVING

Question 3:

What is the purpose of the SQL COALESCE function?

a) To concatenate two strings

b) To convert data types

c) To perform bitwise operations

d) To return the first non-null value in a list

Answer: d) To return the first non-null value in a list

Question 4:

Which SQL statement is used to add a new column to an existing table?

a) ALTER TABLE

b) MODIFY TABLE

c) ADD COLUMN

d) INSERT COLUMN

Answer: a) ALTER TABLE

Question 5:

Which SQL function is used to format date and time values?

a) FORMAT()

b) DATE_FORMAT()

c) TO_CHAR()

d) CONVERT()

Answer: b) DATE_FORMAT()

Question 6:

What is the purpose of the SQL ROW_NUMBER() function?

a) To calculate the average of a column

b) To assign a unique sequential integer to each row in the result set

c) To count the number of rows in a table

d) To sort rows based on a column

Answer: b) To assign a unique sequential integer to each row in the result set

Question 7:

Which SQL statement is used to remove a specific constraint from a table?

a) DELETE CONSTRAINT

b) DROP CONSTRAINT

c) REMOVE CONSTRAINT

d) ALTER CONSTRAINT

Answer: b) DROP CONSTRAINT

Question 8:

What is the purpose of the SQL UNION operator?

a) To combine rows from two different tables

b) To join tables based on common columns

c) To return only unique rows from a result set

d) To sort the result set in ascending order

Answer: a) To combine rows from two different tables

Question 9:

Which SQL data type would you use to store large amounts of text data?

a) VARCHAR

b) TEXT

c) STRING

d) CHAR

Answer: b) TEXT

Question 10:

In a SQL statement, what does the EXISTS operator do?

a) Checks if a column exists in a table

b) Checks if a table exists in the database

c) Checks if a value exists in a column

d) Checks if a subquery returns any rows

Answer: d) Checks if a subquery returns any rows

Question 11:

What is the purpose of the SQL CASE statement?

a) To create a new table in the database

b) To update existing rows in a table

c) To perform calculations on columns

d) To conditionally return a value based on specified conditions

Answer: d) To conditionally return a value based on specified conditions

Question 12:

Which SQL function is used to calculate the total cumulative sum of a column's values?

a) SUM()

b) TOTAL()

c) CUMSUM()

d) ROLLUP()

Answer: c) CUMSUM()

Question 13:

What is the purpose of the SQL PRIMARY KEY constraint?

a) To allow null values in a column

b) To ensure data uniqueness in a column

c) To allow duplicate values in a column

d) To define a default value for a column

Answer: b) To ensure data uniqueness in a column

Question 14:

Which SQL statement is used to change data in multiple rows at once?

a) UPDATE

b) ALTER

c) MODIFY

d) SET

Answer: a) UPDATE

Question 15:

Which SQL clause is used to restrict the number of rows returned by a query based on a specific range?

a) LIMIT

b) OFFSET

c) FETCH

d) ROWS

Answer: b) OFFSET

Question 16:

What is the purpose of the SQL LAG() function?

a) To retrieve the previous row's value in a column

b) To calculate the logarithm of a number

c) To find the lowest value in a column

d) To generate random numbers

Answer: a) To retrieve the previous row's value in a column

Question 17:

Which SQL command is used to create an index on a table's column?

a) ADD INDEX

b) CREATE INDEX

c) INDEX TABLE

d) MAKE INDEX

Answer: b) CREATE INDEX

Question 18:

What is the purpose of the SQL RANK() function?

a) To calculate the median of a column

b) To assign a unique rank to each row in the result set

c) To count the number of rows in a table

d) To group rows based on a column

Answer: b) To assign a unique rank to each row in the result set

Question 19:

Which SQL command is used to delete a table's structure and data?

a) DELETE TABLE

b) REMOVE TABLE

c) DROP TABLE

d) ERASE TABLE

Answer: c) DROP TABLE

Question 20:

What is the purpose of the SQL TRIGGER?

a) To execute a batch of SQL statements

b) To automate the execution of a series of queries

c) To generate random data for testing purposes

d) To perform an action in response to a database event

Answer: d) To perform an action in response to a database event

**Here are 10 more multiple-choice questions based on a hypothetical "College" database schema along with their answers:**

Schema: College Database

Tables:

1. Students (student_id, first_name, last_name, date_of_birth)

2. Courses (course_id, course_name, department, credits)

3. Enrollments (enrollment_id, student_id, course_id, enrollment_date)

Question 1:

Which SQL query would you use to retrieve the names of all students born after January 1, 2000?

a) SELECT first_name, last_name FROM Students WHERE date_of_birth > '2000-01-01';

b) SELECT first_name, last_name FROM Students WHERE date_of_birth < '2000-01-01';

c) SELECT first_name, last_name FROM Students WHERE date_of_birth >= '2000-01-01';

d) SELECT first_name, last_name FROM Students WHERE date_of_birth <= '2000-01-01';

Answer: a) SELECT first_name, last_name FROM Students WHERE date_of_birth > '2000-01-01';

Question 2:

What will the following SQL query do?

```sql
SELECT course_name, COUNT() as enrollment_count
FROM Courses
JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY course_name;
```

a) Retrieve a list of course names and the number of enrollments for each course.

b) Retrieve a list of course names and the total count of courses.

c) Retrieve a list of course names with no enrollments.

d) Retrieve a list of course names and the number of students enrolled in each course.

Answer: a) Retrieve a list of course names and the number of enrollments for each course.

Question 3:

Which SQL statement is used to add a new student to the "Students" table?

a) INSERT INTO Students (student_id, first_name, last_name, date_of_birth) VALUES (1, 'John', 'Doe', '2002-05-15');

b) UPDATE Students SET first_name = 'John', last_name = 'Doe', date_of_birth = '2002-05-15' WHERE student_id = 1;

c) INSERT Students (first_name, last_name, date_of_birth) VALUES ('John', 'Doe', '2002-05-15');

d) ADD INTO Students (student_id, first_name, last_name, date_of_birth) VALUES (1, 'John', 'Doe', '2002-05-15');

Answer: a) INSERT INTO Students (student_id, first_name, last_name, date_of_birth) VALUES (1, 'John', 'Doe', '2002-05-15');

Question 4:

Which SQL query would you use to find the student with the highest number of enrollments?

a) SELECT first_name, last_name FROM Students ORDER BY COUNT() DESC LIMIT 1;

b) SELECT first_name, last_name FROM Students WHERE student_id = MAX(COUNT());

c) SELECT first_name, last_name FROM Students WHERE student_id = (SELECT MAX(COUNT()) FROM Enrollments);

d) SELECT Students.first_name, Students.last_name

```
  FROM Students
  JOIN Enrollments ON Students.student_id = Enrollments.student_id
  GROUP BY Students.student_id
  ORDER BY COUNT() DESC
  LIMIT 1;
```

Answer: d) SELECT Students.first_name, Students.last_name ...

Question 5:

Which SQL query would you use to retrieve the course names that have no enrollments?

a) SELECT course_name FROM Courses WHERE course_id NOT IN (SELECT course_id FROM Enrollments);

b) SELECT course_name FROM Courses LEFT JOIN Enrollments ON Courses.course_id = Enrollments.course_id WHERE Enrollments.course_id IS NULL;

c) SELECT course_name FROM Courses INNER JOIN Enrollments ON Courses.course_id = Enrollments.course_id WHERE Enrollments.course_id IS NULL;

d) SELECT course_name FROM Courses WHERE course_id = (SELECT MAX(course_id) FROM Enrollments);

Answer: b) SELECT course_name FROM Courses LEFT JOIN Enrollments ON ...

Question 6:

What does the following SQL query do?

```sql
SELECT department, AVG(credits) as avg_credits

FROM Courses

GROUP BY department

HAVING AVG(credits) > 3.0;
```

a) Retrieves the average number of credits for courses in each department.

b) Retrieves the departments with an average course credit greater than 3.0.

c) Retrieves all courses and departments with an average credit greater than 3.0.

d) Retrieves all departments and their average course credit, regardless of value.

Answer: b) Retrieves the departments with an average course credit greater than 3.0.

Question 7:

Which SQL statement would you use to remove a student from the "Students" table?

a) DELETE FROM Students WHERE student_id = 1;

b) REMOVE FROM Students WHERE student_id = 1;

c) ERASE FROM Students WHERE student_id = 1;

d) CLEAR FROM Students WHERE student_id = 1;

Answer: a) DELETE FROM Students WHERE student_id = 1;

Question 8:

Which SQL query would you use to find the names of students who are enrolled in more than two courses?

a) SELECT first_name, last_name FROM Students WHERE student_id IN (SELECT student_id FROM Enrollments GROUP BY student_id HAVING COUNT() > 2);

b) SELECT first_name, last_name FROM Students WHERE COUNT() > 2;

c) SELECT first_name, last_name FROM Students WHERE student_id IN (SELECT student_id FROM Enrollments WHERE COUNT() > 2);

d) SELECT first_name, last_name FROM Students WHERE student_id IN (SELECT student_id FROM Enrollments HAVING COUNT() > 2);

Answer: a) SELECT first_name, last_name FROM Students WHERE ...

Question 9:

Which SQL query would you use to update the department of a specific course?

a) UPDATE Courses SET department = 'Mathematics' WHERE course_id = 101;

b) ALTER Courses UPDATE department = 'Mathematics' WHERE course_id = 101;

c) MODIFY Courses SET department = 'Mathematics' WHERE course_id = 101;

d) CHANGE Courses SET department = 'Mathematics' WHERE course_id = 101;

Answer: a) UPDATE Courses SET department = 'Mathematics' WHERE course_id = 101;

Question 10:

What will the following SQL query do?

```sql

SELECT course_name, COUNT(student_id) as enrollment_count
```

```
FROM Courses
LEFT JOIN Enrollments ON Courses.course_id = Enrollments.course_id
GROUP BY course_name;
```

a) Retrieve a list of course names and the number of students enrolled in each course.

b) Retrieve a list of course names and the total count of courses.

c) Retrieve a list of course names with no enrollments.

d) Retrieve a list of course names and the number of enrollments for each course.

Answer: a) Retrieve a list of course names and the number of students enrolled in each course.

Question 11:

Which SQL query would you use to find the student who has enrolled in the most courses?

a) SELECT student_id FROM Enrollments GROUP BY student_id ORDER BY COUNT() DESC LIMIT 1;

b) SELECT student_id FROM Enrollments GROUP BY student_id ORDER BY COUNT() ASC LIMIT 1;

c) SELECT student_id FROM Enrollments ORDER BY COUNT() DESC LIMIT 1;

d) SELECT student_id FROM Enrollments WHERE COUNT() = MAX(COUNT());

Answer: a) SELECT student_id FROM Enrollments GROUP BY ...

Question 12:

What will the following SQL query do?

```sql
SELECT first_name, last_name
FROM Students
WHERE student_id NOT IN (SELECT student_id FROM Enrollments);
```

a) Retrieve the names of students who are not enrolled in any course.

b) Retrieve the names of students who are enrolled in at least one course.

c) Retrieve the names of students who have the highest student_id value.

d) Retrieve the names of students who have enrolled in all available courses.

Answer: a) Retrieve the names of students who are not enrolled in any course.

Question 13:

Which SQL query would you use to retrieve the course names and the total credits of all courses?

a) SELECT course_name, SUM(credits) as total_credits FROM Courses;

b) SELECT course_name, SUM(credits) as total_credits FROM Courses GROUP BY course_name;

c) SELECT course_name, total_credits FROM Courses;

d) SELECT course_name, COUNT(credits) as total_credits FROM Courses;

Answer: b) SELECT course_name, SUM(credits) as total_credits FROM Courses GROUP BY course_name;

**Here are 10 multiple-choice questions based on a hypothetical "Bank" database schema along with their answers:**

Schema: Bank Database

Tables:

1. Customers (customer_id, first_name, last_name, date_of_birth)

2. Accounts (account_id, customer_id, balance, account_type)

3. Transactions (transaction_id, account_id, transaction_date, amount, transaction_type)

Question 1:

Which SQL query would you use to retrieve the total balance of all accounts for a specific customer?

a) SELECT SUM(balance) FROM Accounts WHERE customer_id = 123;

b) SELECT total(balance) FROM Accounts WHERE customer_id = 123;

c) SELECT balance FROM Accounts WHERE customer_id = 123;

d) SELECT SUM(balance) FROM Customers WHERE customer_id = 123;

Answer: a) SELECT SUM(balance) FROM Accounts WHERE customer_id = 123;

Question 2:

What will the following SQL query do?

```sql
SELECT first_name, last_name, COUNT() as transaction_count
FROM Customers
JOIN Accounts ON Customers.customer_id = Accounts.customer_id
JOIN Transactions ON Accounts.account_id = Transactions.account_id
WHERE Transactions.transaction_type = 'Withdrawal'
GROUP BY first_name, last_name;
```

a) Retrieve the names of customers who have made withdrawals and the number of withdrawals each.

b) Retrieve the names of customers who have made deposits and the number of deposits each.

c) Retrieve the names of customers who have made transactions and the total count of transactions.

d) Retrieve the names of customers who have made withdrawals and the total count of withdrawals.

Answer: a) Retrieve the names of customers who have made withdrawals and the number of withdrawals each.

Question 3:

Which SQL statement would you use to add a new customer to the "Customers" table?

a) INSERT INTO Customers (customer_id, first_name, last_name, date_of_birth) VALUES (101, 'John', 'Doe', '1990-03-15');

b) UPDATE Customers SET first_name = 'John', last_name = 'Doe', date_of_birth = '1990-03-15' WHERE customer_id = 101;

c) INSERT Customers (first_name, last_name, date_of_birth) VALUES ('John', 'Doe', '1990-03-15');

d) ADD INTO Customers (customer_id, first_name, last_name, date_of_birth) VALUES (101, 'John', 'Doe', '1990-03-15');

Answer: a) INSERT INTO Customers (customer_id, first_name, last_name, date_of_birth) VALUES (101, 'John', 'Doe', '1990-03-15');

Question 4:

What is the purpose of the SQL JOIN operation?

a) To filter rows based on a condition

b) To group rows based on a column

c) To combine data from two or more tables based on a related column

d) To sort the result set in ascending or descending order

Answer: c) To combine data from two or more tables based on a related column

Question 5:

Which SQL query would you use to find the average balance of accounts for each account type?

a) SELECT AVG(balance), account_type FROM Accounts GROUP BY account_type;

b) SELECT AVG(balance), account_type FROM Customers GROUP BY account_type;

c) SELECT AVG(account_type), balance FROM Accounts GROUP BY account_type;

d) SELECT AVG(balance), account_id FROM Accounts GROUP BY account_type;

Answer: a) SELECT AVG(balance), account_type FROM Accounts GROUP BY account_type;

Question 6:

What will the following SQL query do?

```sql
SELECT account_id, MAX(balance) as highest_balance
FROM Accounts
GROUP BY account_id
HAVING MAX(balance) > 10000;
```

a) Retrieve the account IDs and their highest balances for accounts with balances over 10000.

b) Retrieve the account IDs and their highest balances for all accounts.

c) Retrieve the account IDs and their highest balances for accounts with balances less than 10000.

d) Retrieve the account IDs and the sum of balances for accounts with balances over 10000.

Answer: a) Retrieve the account IDs and their highest balances for accounts with balances over 10000.

Question 7:

Which SQL query would you use to find the account IDs with the most transactions?

a) SELECT account_id FROM Transactions GROUP BY account_id ORDER BY COUNT() DESC LIMIT 1;

b) SELECT account_id FROM Transactions WHERE COUNT() = MAX(COUNT());

c) SELECT account_id FROM Transactions GROUP BY account_id ORDER BY COUNT() ASC LIMIT 1;

d) SELECT account_id FROM Transactions WHERE COUNT() >= AVG(COUNT());

Answer: a) SELECT account_id FROM Transactions GROUP BY ...

Question 8:

What does the SQL ORDER BY clause do?

a) It filters rows based on a condition.

b) It groups rows based on a column.

c) It combines data from multiple tables.

d) It sorts the result set in ascending or descending order.

Answer: d) It sorts the result set in ascending or descending order.

Question 9:

Which SQL query would you use to retrieve the account IDs and the total amount deposited for each account?

a) SELECT account_id, SUM(amount) as total_deposits FROM Transactions WHERE transaction_type = 'Deposit' GROUP BY account_id;

b) SELECT account_id, SUM(amount) as total_deposits FROM Accounts WHERE transaction_type = 'Deposit' GROUP BY account_id;

c) SELECT account_id, SUM(amount) as total_deposits FROM Transactions GROUP BY account_id WHERE transaction_type = 'Deposit';

d) SELECT account_id, SUM(deposit) as total_deposits FROM Transactions GROUP BY account_id;

Answer: a) SELECT account_id, SUM(amount) as total_deposits FROM Transactions WHERE transaction_type = 'Deposit' GROUP BY account_id;

Question 10:

What will the following SQL query do?

```sql
SELECT customer_id, AVG(balance) as average_balance

FROM Customers

JOIN Accounts ON Customers.customer_id = Accounts.customer_id

GROUP BY customer_id

HAVING AVG(balance) > 5000;
```

a) Retrieve the IDs of customers with an average account balance greater than 5000.

b) Retrieve the IDs of customers with a total account balance greater than 5000.

c) Retrieve the IDs of customers with an average account balance less than 5000.

d) Retrieve the IDs of customers with an average transaction amount greater than 5000.

Answer: a) Retrieve the IDs of customers with an average account balance greater than 5000.

**Here are 10 more multiple-choice questions based on a hypothetical "Hospital Management" database schema along with their answers:**

Schema: Hospital Management Database

Tables:

1. Patients (patient_id, first_name, last_name, date_of_birth, gender)

2. Doctors (doctor_id, first_name, last_name, specialization, experience)

3. Appointments (appointment_id, patient_id, doctor_id, appointment_date, status)

4. Medications (medication_id, patient_id, medication_name, dosage)

Question 11:

Which SQL query would you use to retrieve the names of all female patients born after 1990?

a) SELECT first_name, last_name FROM Patients WHERE gender = 'Female' AND date_of_birth > '1990-01-01';

b) SELECT first_name, last_name FROM Patients WHERE gender = 'Female' OR date_of_birth > '1990-01-01';

c) SELECT first_name, last_name FROM Patients WHERE gender = 'Female' AND date_of_birth < '1990-01-01';

d) SELECT first_name, last_name FROM Patients WHERE gender = 'Female' AND date_of_birth > '1990-12-31';

Answer: a) SELECT first_name, last_name FROM Patients WHERE gender = 'Female' AND date_of_birth > '1990-01-01';

Question 12:

What will the following SQL query do?

```sql
SELECT specialization, COUNT() as doctor_count
FROM Doctors
GROUP BY specialization
HAVING COUNT() > 5;
```

a) Retrieve the specialization of doctors with more than 5 years of experience.

b) Retrieve the specialization of doctors with exactly 5 years of experience.

c) Retrieve the specialization of doctors with fewer than 5 years of experience.

d) Retrieve the specialization of doctors with more than 5 doctors in that specialization.

Answer: d) Retrieve the specialization of doctors with more than 5 doctors in that specialization.

Question 13:

Which SQL statement would you use to add a new patient to the "Patients" table?

a) INSERT INTO Patients (patient_id, first_name, last_name, date_of_birth, gender) VALUES (101, 'Alice', 'Johnson', '1985-08-15', 'Female');

b) UPDATE Patients SET first_name = 'Alice', last_name = 'Johnson', date_of_birth = '1985-08-15', gender = 'Female' WHERE patient_id = 101;

c) INSERT Patients (first_name, last_name, date_of_birth, gender) VALUES ('Alice', 'Johnson', '1985-08-15', 'Female');

d) ADD INTO Patients (patient_id, first_name, last_name, date_of_birth, gender) VALUES (101, 'Alice', 'Johnson', '1985-08-15', 'Female');

Answer: a) INSERT INTO Patients (patient_id, first_name, last_name, date_of_birth, gender) VALUES (101, 'Alice', 'Johnson', '1985-08-15', 'Female');

Question 14:

What is the purpose of the SQL WHERE clause?

a) To group rows based on a condition

b) To filter rows based on a condition

c) To sort the result set in ascending or descending order

d) To join multiple tables

Answer: b) To filter rows based on a condition

Question 15:

Which SQL query would you use to find the average experience of doctors in each specialization?

a) SELECT AVG(experience), specialization FROM Doctors GROUP BY specialization;

b) SELECT specialization, AVG(experience) FROM Doctors GROUP BY specialization;

c) SELECT AVG(specialization), experience FROM Doctors GROUP BY specialization;

d) SELECT AVG(experience), doctor_id FROM Doctors GROUP BY specialization;

Answer: a) SELECT AVG(experience), specialization FROM Doctors GROUP BY specialization;

Question 16:

What does the SQL GROUP BY clause do?

a) It combines data from multiple tables.

b) It sorts the result set in ascending or descending order.

c) It filters rows based on a condition.

d) It groups rows based on a column.

Answer: d) It groups rows based on a column.

Question 17:

Which SQL query would you use to retrieve the names of patients who have appointments?

a) SELECT first_name, last_name FROM Patients JOIN Appointments ON Patients.patient_id = Appointments.patient_id;

b) SELECT first_name, last_name FROM Patients WHERE patient_id IN (SELECT patient_id FROM Appointments);

c) SELECT first_name, last_name FROM Patients WHERE patient_id NOT IN (SELECT patient_id FROM Appointments);

d) SELECT first_name, last_name FROM Patients JOIN Appointments WHERE Patients.patient_id = Appointments.patient_id;

Answer: b) SELECT first_name, last_name FROM Patients WHERE patient_id IN (SELECT patient_id FROM Appointments);

Question 18:

Which SQL query would you use to find the patient who has the most appointments?

a) SELECT patient_id FROM Appointments GROUP BY patient_id ORDER BY COUNT() DESC LIMIT 1;

b) SELECT patient_id FROM Appointments WHERE COUNT() = MAX(COUNT());

c) SELECT patient_id FROM Appointments GROUP BY patient_id ORDER BY COUNT() ASC LIMIT 1;

d) SELECT patient_id FROM Appointments WHERE COUNT() >= AVG(COUNT());

Answer: a) SELECT patient_id FROM Appointments GROUP BY patient_id ...

Question 19:

What does the SQL JOIN operation do?

a) It filters rows based on a condition.

b) It groups rows based on a column.

c) It combines data from two or more tables based on a related column.

d) It sorts the result set in ascending or descending order.

Answer: c) It combines data from two or more tables based on a related column.

Question 20:

Which SQL query would you use to retrieve the doctor names and the total number of appointments for each doctor?

a) SELECT first_name, last_name, COUNT(appointment_id) as total_appointments FROM Doctors JOIN Appointments ON Doctors.doctor_id = Appointments.doctor_id GROUP BY first_name, last_name;

b) SELECT first_name, last_name, COUNT(appointment_id) as total_appointments FROM Appointments GROUP BY Doctors.doctor_id;

c) SELECT first_name, last_name, COUNT() as total_appointments FROM Doctors JOIN Appointments ON Doctors.doctor_id = Appointments.doctor_id GROUP BY Doctors.doctor_id;

d) SELECT first_name, last_name, COUNT(appointment_id) as total_appointments FROM Appointments GROUP BY first_name, last_name;

Answer: a) SELECT first_name, last_name, COUNT(appointment_id) as total_appointments ...

**Here are 3 multiple-choice questions based on a hypothetical "Airline" database schema along with their answers:**

Schema: Airline Database

Tables:

1. Flights (flight_id, flight_number, departure_city, arrival_city, departure_time, arrival_time)

2. Passengers (passenger_id, first_name, last_name, gender, date_of_birth)

3. Bookings (booking_id, flight_id, passenger_id, booking_date, seat_number)

4. Aircrafts (aircraft_id, aircraft_type, total_seats)

Question 1:

Which SQL query would you use to retrieve the list of passengers who have booked flights departing from "New York"?

a) SELECT first_name, last_name FROM Passengers WHERE passenger_id IN (SELECT passenger_id FROM Bookings WHERE flight_id IN (SELECT flight_id FROM Flights WHERE departure_city = 'New York'));

b) SELECT first_name, last_name FROM Passengers WHERE passenger_id IN (SELECT passenger_id FROM Bookings WHERE flight_id IN (SELECT flight_id FROM Flights WHERE arrival_city = 'New York'));

c) SELECT first_name, last_name FROM Passengers WHERE passenger_id IN (SELECT passenger_id FROM Bookings WHERE flight_id IN (SELECT flight_id FROM Flights WHERE arrival_city = 'New York'));

d) SELECT first_name, last_name FROM Passengers WHERE passenger_id IN (SELECT passenger_id FROM Bookings WHERE flight_id IN (SELECT flight_id FROM Flights WHERE departure_city = 'New York'));

Answer: a) SELECT first_name, last_name FROM Passengers WHERE passenger_id IN ...

Question 2:

What will the following SQL query do?

```sql
SELECT flight_number, COUNT() as passenger_count

FROM Flights

JOIN Bookings ON Flights.flight_id = Bookings.flight_id

GROUP BY flight_number

HAVING COUNT() > 100;
```

a) Retrieve the flight numbers of flights with more than 100 passengers.

b) Retrieve the flight numbers of flights with exactly 100 passengers.

c) Retrieve the flight numbers of flights with fewer than 100 passengers.

d) Retrieve the flight numbers of flights with an average of more than 100 passengers.

Answer: a) Retrieve the flight numbers of flights with more than 100 passengers.

Question 3:

Which SQL statement would you use to add a new flight to the "Flights" table?

a) INSERT INTO Flights (flight_id, flight_number, departure_city, arrival_city, departure_time, arrival_time) VALUES (101, 'AB123', 'Los Angeles', 'New York', '2023-09-01 08:00:00', '2023-09-01 15:00:00');

b) UPDATE Flights SET flight_number = 'AB123', departure_city = 'Los Angeles', arrival_city = 'New York', departure_time = '2023-09-01 08:00:00', arrival_time = '2023-09-01 15:00:00' WHERE flight_id = 101;

c) INSERT Flights (flight_number, departure_city, arrival_city, departure_time, arrival_time) VALUES ('AB123', 'Los Angeles', 'New York', '2023-09-01 08:00:00', '2023-09-01 15:00:00');

d) ADD INTO Flights (flight_id, flight_number, departure_city, arrival_city, departure_time, arrival_time) VALUES (101, 'AB123', 'Los Angeles', 'New York', '2023-09-01 08:00:00', '2023-09-01 15:00:00');

Answer: a) INSERT INTO Flights (flight_id, flight_number, departure_city, arrival_city, departure_time, arrival_time) VALUES (101, 'AB123', 'Los Angeles', 'New York', '2023-09-01 08:00:00', '2023-09-01 15:00:00');

**here are 5 multiple-choice questions based on a simplified "Bank" database schema along with sample table data. Please note that this is a simplified schema and sample data for illustrative purposes:**

Schema: Bank Database

Tables:

1. Customers (customer_id, first_name, last_name, date_of_birth, balance)

2. Accounts (account_id, customer_id, account_type, balance)

3. Transactions (transaction_id, account_id, transaction_date, amount, transaction_type)

Sample Data:

Customers Table

| customer_id | first_name | last_name | date_of_birth | balance |

| | | | | |
|---|---|---|---|---|
| 101 | John | Doe | 1990-05-15 | 5000 |
| 102 | Jane | Smith | 1985-09-20 | 8000 |
| 103 | Michael | Johnson | 1978-03-10 | 12000 |

Accounts Table

| account_id | customer_id | account_type | balance |
|---|---|---|---|
| 201 | 101 | Savings | 3000 |
| 202 | 101 | Checking | 2000 |
| 203 | 102 | Savings | 5000 |

Transactions Table

| transaction_id | account_id | transaction_date | amount | transaction_type |
|---|---|---|---|---|
| 301 | 201 | 2023-08-01 10:00:00 | 500 | Deposit |
| 302 | 202 | 2023-08-02 14:30:00 | 1000 | Withdrawal |
| 303 | 203 | 2023-08-03 09:15:00 | 200 | Deposit |

Question 1:

Which SQL query would you use to retrieve the first name and last name of customers with a balance greater than 6000?

a) SELECT first_name, last_name FROM Customers WHERE balance > 6000;

b) SELECT first_name, last_name FROM Customers WHERE balance >= 6000;

c) SELECT first_name, last_name FROM Customers HAVING balance > 6000;

d) SELECT first_name, last_name FROM Customers GROUP BY balance > 6000;

Answer: a) SELECT first_name, last_name FROM Customers WHERE balance > 6000;

Question 2:

What will the following SQL query do?

```sql
SELECT account_type, COUNT() as account_count
FROM Accounts
GROUP BY account_type
HAVING COUNT() > 1;
```

a) Retrieve the count of accounts for each account type.

b) Retrieve the account types with more than one account.

c) Retrieve the account types with only one account.

d) Retrieve the account types with exactly one account.

Answer: b) Retrieve the account types with more than one account.

Question 3:

Which SQL statement would you use to add a new transaction to the "Transactions" table?

a) INSERT INTO Transactions (transaction_id, account_id, transaction_date, amount, transaction_type) VALUES (401, 201, '2023-08-05 11:30:00', 300, 'Deposit');

b) UPDATE Transactions SET transaction_id = 401, account_id = 201, transaction_date = '2023-08-05 11:30:00', amount = 300, transaction_type = 'Deposit' WHERE transaction_id = 401;

c) INSERT Transactions (account_id, transaction_date, amount, transaction_type) VALUES (201, '2023-08-05 11:30:00', 300, 'Deposit');

d) ADD INTO Transactions (transaction_id, account_id, transaction_date, amount, transaction_type) VALUES (401, 201, '2023-08-05 11:30:00', 300, 'Deposit');

Answer: a) INSERT INTO Transactions (transaction_id, account_id, transaction_date, amount, transaction_type) VALUES (401, 201, '2023-08-05 11:30:00', 300, 'Deposit');

Question 4:

What is the purpose of the SQL GROUP BY clause?

a) To combine data from multiple tables.

b) To sort the result set in ascending or descending order.

c) To filter rows based on a condition.

d) To group rows based on a column.

Answer: d) To group rows based on a column.

Question 5:

Which SQL query would you use to find the average balance of customers for each account type?

a) SELECT AVG(balance), account_type FROM Customers GROUP BY account_type;

b) SELECT AVG(balance), account_type FROM Accounts GROUP BY account_type;

c) SELECT AVG(account_type), balance FROM Customers GROUP BY account_type;

d) SELECT AVG(balance), customer_id FROM Customers GROUP BY account_type;

Answer: b) SELECT AVG(balance), account_type FROM Accounts GROUP BY account_type;