```

Name :

Role :

Date of Task :

Time Taken:

```

# SQL - L1(Level-1)

## QDLC

1. Who is often credited with inventing SQL?

2. In what year was SQL first developed?

3. What company developed the first version of SQL?

4. What was the original name of SQL?

5. What programming language influenced the creation of SQL?

6. When was the first commercial implementation of SQL released?

7. What was the initial purpose of SQL when it was created?

8. What were some of the early versions of SQL called?

9. In which decade did SQL become a standard?

10. What organization standardized SQL?

11. When was SQL first standardized?

12. What are the three major categories of SQL statements?

13. What was the primary motivation behind creating SQL?

14. How has SQL evolved over the years?

15. What are some of the key features introduced in SQL-92?

16. What role did IBM play in the development of SQL?

17. When did SQL start gaining popularity outside of academia and research?

18. What are the differences between SQL and other query languages of its time?

19. Who were some of the early contributors to the development of SQL?

20. What are some of the notable SQL-based database management systems that emerged in the 1970s and 1980s?

## ADLC

\* Answer should be here:

Ans 1.

SQL, which stands for Structured Query Language, was not invented by a single individual. Instead, it was developed collaboratively over several years by a group of IBM researchers led by Donald D. Chamberlin and Raymond F. Boyce. In the early 1970s, they published a paper titled "SEQUEL: A Structured English Query Language" that described the fundamentals of SQL.

Ans 2.

SQL, or Structured Query Language, was first developed in the early 1970s.

Ans 3.

The first version of SQL (Structured Query Language) was developed by IBM. SQL originated as a project within IBM's research division in the early 1970s.

Ans 4.

The original name of SQL was not "SQL" but rather "SEQUEL," which stands for "Structured English Query Language." SQL was initially developed as part of a project at IBM in the early 1970s, and it was referred to as SEQUEL in the early stages

Ans 5.

The creation of SQL (Structured Query Language) was influenced by several programming languages and systems. One of the primary influences on SQL was a language called "Datalog." Datalog is a declarative query language that was developed for deductive database systems.

Ans 6.

The first commercial implementation of SQL was released by the company Relational Software, Inc., which later became Oracle Corporation. They released their commercial RDBMS (Relational Database Management System) called Oracle RDBMS in 1979. This was one of the earliest RDBMS products available commercially that supported SQL as its query language.

Ans 7.

The initial purpose of SQL (Structured Query Language) when it was created was to provide a standardized and user-friendly way to interact with and query data stored in relational database management systems (RDBMS). SQL was developed at IBM in the early 1970s as part of the System R project, and its primary goal was to address the growing need for a more efficient and accessible method for working with data in relational databases.

Ans 8.

Here are some of the early versions or SQL-related systems:

1.SEQUEL

2.System R SQL

3.Ingres

4.PostgreSQL

5.Informix

6.Oracle

Ans 9.

SQL became a recognized and standardized language for relational databases in the 1980s, with ongoing efforts to refine and expand the standard over the subsequent decades. The standardization process helped ensure compatibility and portability of SQL across different database management systems.

Ans 10.

The standardization of SQL (Structured Query Language) was primarily undertaken by two organizations:

1. **American National Standards Institute (ANSI)**
2. **International Organization for Standardization (ISO)**

Ans 11.

SQL (Structured Query Language) was first standardized in 1986 with the publication of the SQL-86 standard by the American National Standards Institute (ANSI). This was one of the earliest efforts to standardize SQL and establish it as a common language for relational database management systems (RDBMS).

Ans 12.

SQL statements can be broadly categorized into three major categories:

1. **Data Query Language (DQL) Statements**
2. **Data Manipulation Language (DML) Statements**
3. **Data Definition Language (DDL) Statements**

**Ans 13.**

The primary motivation behind creating SQL (Structured Query Language) was to provide a standardized and user-friendly way to interact with and manage data stored in relational database management systems (RDBMS). Several factors contributed to the development of SQL:

1.Data Retrieval and Manipulation

2.Data Independence

3.Standardization

4.Ease of Use

5.Efficiency

Ans 14.

SQL (Structured Query Language) has evolved significantly since its inception in the 1970s. Its evolution has been driven by several factors, including technological advancements, changing database needs, and efforts to standardize and enhance the language. Here's a brief overview of how SQL has evolved over the years:

1. **Early SQL Dialects (1970s and 1980s)**:
   * SQL began as a research project at IBM, and early implementations had variations in syntax and functionality.
   * Different database vendors introduced their own SQL dialects, leading to some incompatibilities and vendor lock-in.
2. **SQL Standardization (1980s)**:
   * ANSI and ISO started standardizing SQL to provide a common language across different database systems.
   * SQL-86 was the first official SQL standard, followed by SQL-89 and SQL-92.
   * These standards established core SQL features and syntax, making SQL more portable across databases.
3. **Expansion of SQL Standards (1990s and 2000s)**:
   * SQL:1999 introduced significant enhancements, including support for recursive queries, window functions, and user-defined types.
   * SQL:2003 added features like XML support and improved support for large objects (BLOBs/CLOBs).
   * SQL:2008 introduced more advanced analytics, security features, and temporal data support.
4. **Adoption of SQL in Web Development (2000s)**:
   * With the rise of web applications and e-commerce, SQL became a fundamental language for backend databases.
   * Object-Relational Mapping (ORM) frameworks like Hibernate and Django ORM made it easier to work with SQL databases from high-level programming languages.
5. **NoSQL and New SQL Databases (2010s)**:
   * The emergence of NoSQL databases challenged the dominance of traditional SQL databases in certain use cases.
   * New SQL databases, like NewSQL and in-memory databases, aimed to combine the benefits of SQL with scalability and performance.
6. **SQL:2011 and Beyond**:
   * SQL:2011 continued to refine and extend SQL standards.
   * Modern SQL databases and database management systems (DBMS) offer advanced features like JSON support, geospatial capabilities, and support for big data analytics.
7. **Open Source SQL Databases (Ongoing)**:
   * Open-source SQL databases like PostgreSQL and MySQL have gained popularity and continue to evolve with community contributions.
   * Cloud-based databases, such as Amazon Aurora and Google Cloud SQL, offer managed SQL databases with high availability and scalability.
8. **Big Data and Distributed SQL (Ongoing)**:
   * SQL has adapted to the big data era with technologies like Apache Hive and Presto, enabling SQL queries on massive datasets.
   * Distributed SQL databases have emerged to handle large-scale, distributed data processing while providing SQL interfaces.

SQL's ongoing evolution reflects its adaptability and enduring relevance in the world of data management and analytics. It continues to be a critical tool for working with structured data in various domains, from traditional relational databases to big data and distributed systems.

Ans 15.

Top of Form

SQL-92, also known as SQL2, was a significant milestone in the standardization of SQL (Structured Query Language). It introduced several key features and improvements over its predecessor SQL-89. Here are some of the key features introduced in SQL-92:

1. **Outer Joins**: SQL-92 introduced support for outer joins, allowing you to retrieve rows from one table even if there are no matching rows in another table. This is accomplished using the **LEFT OUTER JOIN**, **RIGHT OUTER JOIN**, and **FULL OUTER JOIN** clauses.
2. **Referential Integrity**: SQL-92 enhanced support for referential integrity constraints, including the ability to define and enforce foreign key constraints between tables. This ensures the integrity of relationships between tables in a relational database.
3. **Views**: SQL-92 introduced the ability to create views, which are virtual tables defined by SQL queries. Views provide an abstraction layer for users and can simplify complex queries by encapsulating them in a reusable view.
4. **Dynamic SQL**: SQL-92 introduced dynamic SQL, allowing you to construct and execute SQL statements dynamically at runtime. This feature is particularly useful for building flexible and customizable database applications.
5. **Common Table Expressions (CTEs)**: SQL-92 introduced Common Table Expressions, which provide a way to define temporary result sets within a SQL query. CTEs improve query readability and maintainability.
6. **Information Schema**: SQL-92 standardized the concept of an information schema, which is a set of read-only system tables and views that provide metadata about the database's structure, such as tables, columns, indexes, and constraints.
7. **User-Defined Data Types (UDTs)**: SQL-92 introduced support for user-defined data types, allowing database designers to define custom data types that can be used in table definitions.
8. **Catalog and Schema**: SQL-92 introduced the concepts of catalog and schema, which help organize database objects. A catalog is a container for schemas, and schemas contain database objects. This feature enhances database organization and management.
9. **Data Types**: SQL-92 expanded the list of data types, including support for date and time data types, INTERVAL data types, and BOOLEAN data types.
10. **Enhanced GROUP BY and HAVING Clauses**: SQL-92 improved the GROUP BY clause by allowing you to include non-aggregated columns in the SELECT clause, making it easier to write complex queries. The HAVING clause was also enhanced to filter grouped results.

These are some of the key features introduced in SQL-92. This standardization played a crucial role in making SQL a more powerful and standardized language for working with relational databases. Many of these features have become integral parts of modern SQL database systems and have been further extended in subsequent SQL standards.

Ans 16.

IBM played a significant role in the development of SQL (Structured Query Language). SQL is a domain-specific language used for managing and querying relational databases, and it was developed at IBM in the early 1970s. Here's a brief overview of IBM's contributions to SQL:

1. IBM Research: SQL was developed at IBM's San Jose Research Laboratory (now known as IBM Almaden Research Center) in the early 1970s. It was initially created as a research project led by Edgar F. Codd, a computer scientist at IBM. Codd's work on relational databases laid the foundation for SQL.
2. System R Project: The development of SQL was closely tied to IBM's System R project. System R was an experimental database management system designed to demonstrate the feasibility of the relational model for databases. SQL was created as the query language for System R.
3. Standardization Efforts: In the late 1970s and early 1980s, IBM played a leading role in standardizing SQL. SQL became an ANSI (American National Standards Institute) standard in 1986 and an ISO (International Organization for Standardization) standard in 1987. IBM's involvement in these standardization efforts helped SQL become widely adopted as the standard language for relational databases.
4. Commercialization: IBM also played a significant role in popularizing SQL by incorporating it into their database management systems, such as DB2. IBM's DB2 database, which was introduced in the 1980s, used SQL as its query language, making SQL accessible to a wide range of users and organizations.
5. Ongoing Development: IBM continued to contribute to the development of SQL by enhancing the language and its database products. Over the years, SQL has evolved and expanded to include various features and capabilities, many of which were influenced by IBM's work in the database field.

In summary, IBM played a pivotal role in the development and standardization of SQL, and their contributions helped establish SQL as the de facto query language for relational databases. SQL has since become a fundamental part of database management and is used by numerous database systems and software applications worldwide.

Top of Form

Ans 17.

SQL started gaining popularity outside of academia and research in the 1980s. While it was initially developed and used for research purposes at IBM's San Jose Research Laboratory in the early 1970s, it took some time for SQL to transition from an experimental language to a widely adopted standard for managing and querying relational databases.

Here are some key factors and events that contributed to SQL's popularity outside of academia and research:

1. Commercial Database Systems: The commercialization of relational database management systems (RDBMS) played a significant role in SQL's adoption. Companies like IBM (with DB2), Oracle (with Oracle Database), and Microsoft (with SQL Server) began offering RDBMS products that used SQL as their query language. These products were marketed to businesses and organizations, leading to increased usage of SQL in practical applications.
2. ANSI and ISO Standardization: The standardization of SQL by ANSI (American National Standards Institute) in 1986 and ISO (International Organization for Standardization) in 1987 gave SQL credibility and made it an industry-standard language. This standardization process provided a common foundation for RDBMS vendors to adhere to, making it easier for developers and users to work with SQL across different database systems.
3. SQL Access in Application Development: SQL's integration into programming languages and application development tools contributed to its popularity. Developers could embed SQL queries directly into their applications, making it easier to interact with databases programmatically. This integration made SQL an essential part of application development.
4. Training and Education: As SQL gained prominence, educational institutions and training programs began incorporating SQL into their curricula. This helped produce a workforce with SQL skills, further fueling its adoption in the industry.
5. Industry Demand: The growing demand for data management solutions, driven by businesses' increasing reliance on data, led to the adoption of SQL-based RDBMS. SQL's ability to handle data efficiently and provide powerful querying capabilities made it a valuable tool for organizations of all sizes.

By the mid-1980s, SQL had become a mainstream technology used in various industries, including finance, healthcare, manufacturing, and more. Its popularity continued to grow throughout the 1990s and beyond, as the use of relational databases and data-driven applications became even more pervasive in the business world. Today, SQL remains a fundamental skill for anyone working with databases and data management.

Ans 18.

SQL (Structured Query Language) stood out from other query languages of its time, primarily due to its success in becoming the industry standard for relational database management systems (RDBMS). Here are some key differences between SQL and other query languages of its time:

1. **Standardization**: One of the most significant differences is SQL's standardization. SQL was adopted as an ANSI (American National Standards Institute) standard in 1986 and later as an ISO (International Organization for Standardization) standard in 1987. This standardization provided a consistent and universal query language for RDBMS, making it easier for developers to write portable code across different database systems.
2. **Declarative Nature**: SQL is a declarative language, which means users specify what they want to retrieve or manipulate from the database, rather than specifying how to do it. This declarative approach abstracted the underlying complexity of database operations and made SQL more user-friendly compared to procedural query languages.
3. **Set-Based Operations**: SQL introduced set-based operations, allowing users to work with data in sets rather than processing individual records sequentially. This set-based approach is highly efficient and aligns with the principles of the relational model.
4. **Integration with Relational Model**: SQL was specifically designed to work with the relational model of data, which was a departure from some earlier query languages that used hierarchical or network models. The relational model's simplicity and elegance appealed to developers and database designers.
5. **Wide Adoption**: SQL gained widespread adoption due to its integration with commercial RDBMS products like IBM's DB2, Oracle Database, and Microsoft SQL Server. These products used SQL as their query language, contributing to its popularity.
6. **Query Optimization**: SQL-based DBMS often had query optimizers that could analyze SQL queries and determine the most efficient way to execute them. This automated optimization made SQL-based systems more efficient and scalable.
7. **Structured Query Language**: The name itself, Structured Query Language, emphasized the structured and organized nature of the language, which appealed to developers and businesses looking for a reliable and consistent way to work with data.
8. **Industry Backing**: SQL received strong backing from prominent companies like IBM, Oracle, and Microsoft, which further solidified its position as the go-to query language for RDBMS.
9. **Standard Language Features**: SQL offered a comprehensive set of features for data retrieval, manipulation, and management. It supported operations like SELECT, INSERT, UPDATE, DELETE, and CREATE TABLE, along with powerful features for data filtering, sorting, and aggregation.
10. **Portability**: SQL's standardization and wide adoption made it possible to write SQL code that could run on various RDBMS without major modifications, reducing vendor lock-in.

While SQL had its advantages, it also faced competition from other query languages, especially in the early days of the relational database era. However, SQL's standardization, broad industry support, and alignment with the relational model ultimately set it apart and contributed to its dominance as the standard query language for RDBMS.

Top of Form

Ans 19.

The development of SQL (Structured Query Language) involved several early contributors who played significant roles in its creation and evolution. Here are some of the key individuals and their contributions:

1. Edgar F. Codd: Edgar F. Codd, a computer scientist at IBM, is often regarded as the father of the relational database model. His 1970 paper titled "A Relational Model of Data for Large Shared Data Banks" laid the theoretical foundation for relational databases and introduced many of the concepts that SQL is built upon. Codd's work at IBM's San Jose Research Laboratory was instrumental in the early development of SQL.
2. Donald D. Chamberlin: Donald D. Chamberlin, also an IBM researcher, worked alongside Raymond F. Boyce to develop a language for querying relational databases. They created SEQUEL (Structured English Query Language), which later became SQL (Structured Query Language). Chamberlin played a pivotal role in the design and implementation of SQL.
3. Raymond F. Boyce: Raymond F. Boyce, another IBM researcher, collaborated with Donald Chamberlin on the SEQUEL project. Their work resulted in the creation of a query language that could efficiently manipulate relational databases. Boyce's contributions were integral to SQL's development.
4. Larry Ellison: While not directly involved in the initial development of SQL, Larry Ellison played a significant role in popularizing SQL through the creation of Oracle Corporation in 1977. Oracle Database, which used SQL as its query language, became one of the most widely used relational database management systems in the world. Ellison's efforts contributed to SQL's adoption in the business world.
5. Chris Date: Chris Date is a renowned database theorist and author who made substantial contributions to the understanding and formalization of relational databases. His books and writings on database theory and SQL have been influential in shaping best practices for SQL usage.
6. IBM Research Team: Beyond the key individuals mentioned, the broader IBM research team at the San Jose Research Laboratory played a vital role in the development of SQL. Their collaborative efforts helped turn the relational model into a practical reality and laid the groundwork for SQL's standardization.

These early contributors collectively shaped the concepts, principles, and language constructs that make up SQL. Their work was crucial in transforming SQL from a research project into a practical and widely adopted tool for managing and querying relational databases.

Top of Form

Ans 20.

The 1970s and 1980s saw the emergence of several notable SQL-based database management systems (DBMS) as the relational database model gained prominence. These systems played a significant role in shaping the database industry and are still used today in various forms. Here are some of the most notable ones:

1. **IBM System R (late 1970s)**: IBM's System R was one of the earliest relational DBMS projects. SQL was initially developed as the query language for System R. Although System R itself didn't become a widely used commercial product, it laid the foundation for the development of SQL and influenced subsequent relational database systems.
2. **Oracle (late 1970s)**: Founded by Larry Ellison in 1977, Oracle Corporation introduced the Oracle Database in the late 1970s. Oracle Database became one of the most prominent SQL-based DBMS in the world and remains a major player in the database industry.
3. **Ingres (mid-1970s)**: Ingres, developed at the University of California, Berkeley, by Michael Stonebraker and his team, was one of the first relational DBMS to use SQL. It was widely used in academic and research settings and influenced the development of other database systems.
4. **PostgreSQL (formerly Postgres, 1980s)**: PostgreSQL, initially known as Postgres, was developed at the University of California, Berkeley, by Michael Stonebraker and his team as an extension of the Ingres project. It later evolved into an open-source SQL database system known for its extensibility and features.
5. **Microsoft SQL Server (1980s)**: Microsoft introduced SQL Server in the late 1980s. It initially competed with Sybase in the SQL Server market. Over time, Microsoft SQL Server became a widely used database management system, particularly in the Windows ecosystem.
6. **Sybase (1980s)**: Sybase, founded in the early 1980s, developed the SQL Server product. Sybase SQL Server was one of the earliest SQL-based DBMS to gain popularity and was used extensively in the financial industry.
7. **Informix (1980s)**: Informix, founded in the early 1980s, introduced its Informix Database, which used SQL as the query language. Informix became known for its performance and scalability.
8. **DB2 (1980s)**: IBM introduced DB2 in the early 1980s. DB2 was one of the first commercial relational DBMS products to use SQL and became a significant player in the enterprise database market.

These SQL-based DBMS, developed in the 1970s and 1980s, paved the way for the widespread adoption of the relational database model and SQL as the standard query language for working with relational databases. Many of these systems are still in use today, and their legacy continues to influence modern database technology.