HARCOURT BUTLER TECHNICAL UNIVERSITY

DEPARTMENT OF MATHEMATICS

SEMINAR PRESENTATION ON

TOPIC : STRUCTURED QUERY LANGUAGE

PRESENTED BY :
ANKIT SINGH
ROLL NO : 220227003

TABLE OF CONTENT

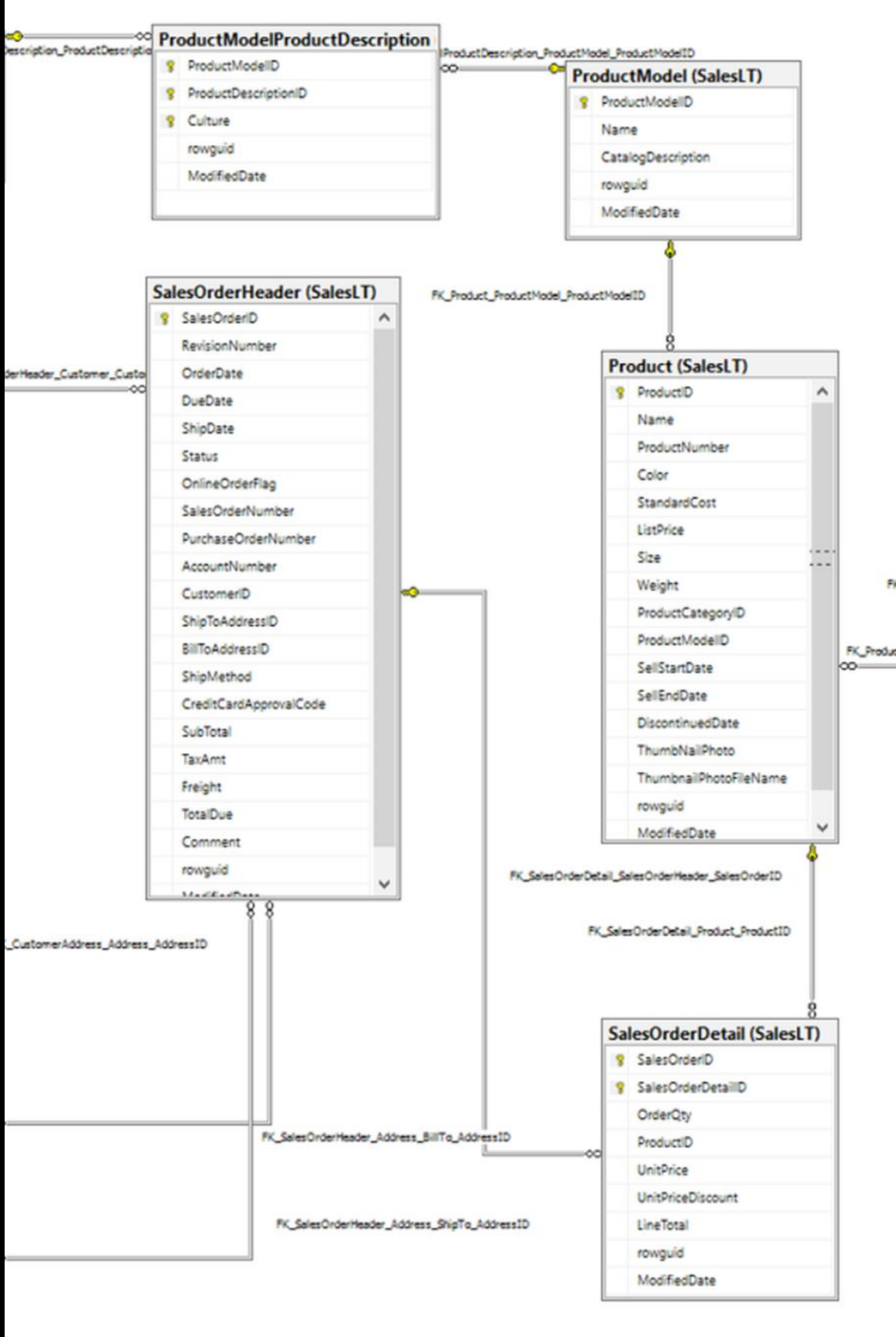
- ✓ Introduction to SQL and MySQL
- ✓ What is Database and its types
- ✓ Characteristics of Databases
- ✓ What is SQL
- ✓ Uses of SQL
- ✓ Basics of SQL Syntax
- ✓ Keypoints of SQL Syntax
- ✓ Statements used in SQL
- ✓ Filtering Data in SQL
- ✓ Real life Implementation in SQL
- ✓ Examples in SQL
- ✓ Pros.
- ✓ Cons.
- ✓ Conclusion
- ✓ References

Introduction to SQL & MySQL

✓ In today's data-driven world, SQL is a foundational for anyone working with data.

✓ It allows users to interact with databases, manipulate data, and extract meaningful insights.

✓ Whether you're a data analyst, data scientist, or a business professional, understanding SQL is essential for effectively working with data.



WHAT IS DATABASE?

- ✓ database is an organized collection of structured information or data stored electronically in a computer system.

- ✓ It's designed to efficiently manage, store, retrieve, and manipulate data according to predefined requirements.

- ✓ Databases serve as repositories for various types of information, allowing users and applications to access, update, and manage data.

Types of Databases

1.Relational Databases: Organize data into tables with predefined relationships between them, following the relational model.
Examples include MySQL, PostgreSQL, SQL Server.

2.NoSQL Databases: Handle unstructured or semi-structured data, providing flexible schema design and scalability.
Examples include MongoDB, Cassandra, Redis.

3.Graph Databases: Designed for managing highly interconnected data, utilizing graph structures to represent relationships between data entities.
Examples include Neo4j, Amazon Neptune.

Characteristics of Databases

1.**Structured Storage:** Data is organized in a structured format, typically using tables, rows, and columns, enabling efficient retrieval and manipulation.

2.**Data Integrity:** Databases enforce rules and constraints to maintain accuracy and consistency of data, ensuring its reliability.

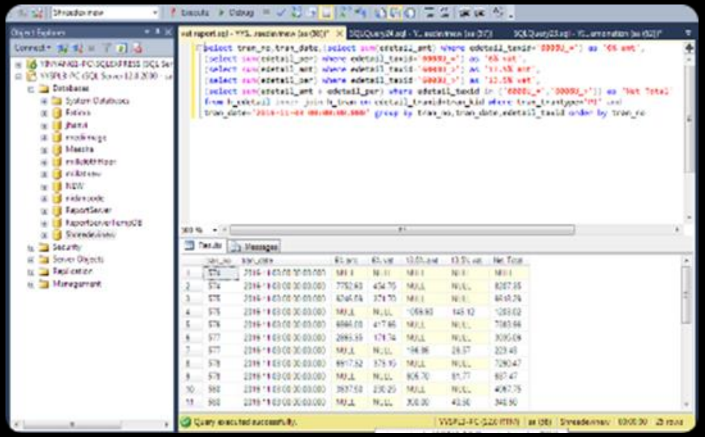
3.**Security:** Databases provide mechanisms to control access, authentication, and authorization, safeguarding sensitive information.

4.**Scalability:** Databases can scale vertically (by adding resources to a single server) or horizontally (by distributing data across multiple servers) to accommodate growing data volumes

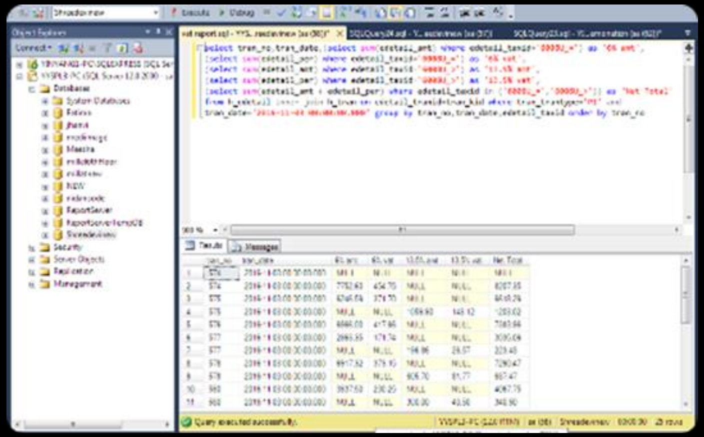
WHAT IS SQL?

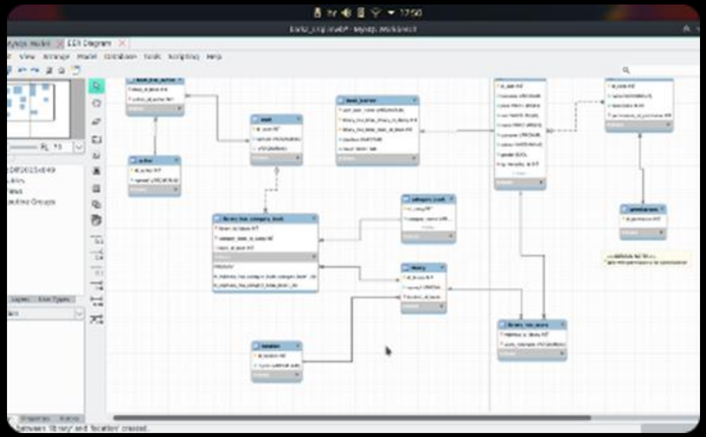
- ✓ SQL is a programming language used for managing and manipulating relational databases.
- ✓ It stands for Structured Query Language.
- ✓ It is used to create, modify, and retrieve data from databases.

Uses of SQL



```
select tran_no,tran_date,(select sum(edetail_amt) where edetail_taxid='00002_+') as '6% amt',
 (select sum(edetail_per) where edetail_taxid='00002_+') as '6% vat',
 (select sum(edetail_amt) where edetail_taxid='00003_>') as '12.5% amt',
 (select sum(edetail_per) where edetail_taxid='00003_>') as '12.5% vat',
 (select sum(edetail_amt + edetail_per) where edetail_taxid in ('00002_+','00003_>')) as 'Net Total'
 from h_edetail inner join h_tran on edetail_tranid=tran_kid where tran_trantype='M' and
 tran_date='2016-11-04 00:00:00.000' group by tran_no,tran_date,edetail_taxid order by tran_no
```





Data Management

SQL is designed to handle large volumes of data efficiently, providing means for storage, retrieval, and manipulation of structured information.

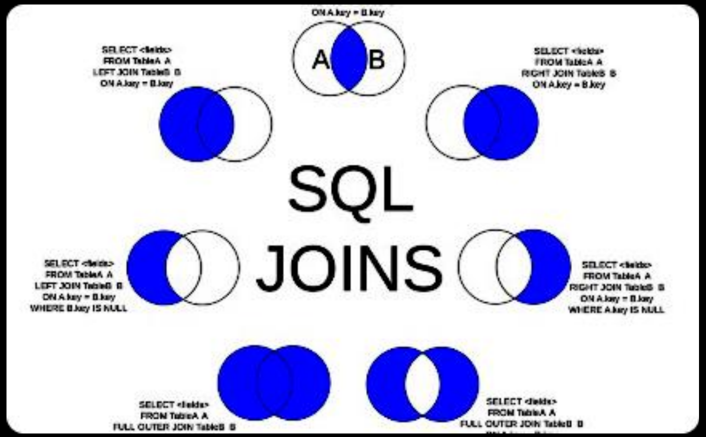
Query Language

It is a standard language for accessing and managing databases, allowing users to to perform tasks such as updating data, running queries, queries, and creating reports. reports.

Database connectivity

SQL facilitates seamless communication with database servers, enabling enabling applications to to interact with the stored stored data effectively.

Basics of SQL syntax



```
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
```

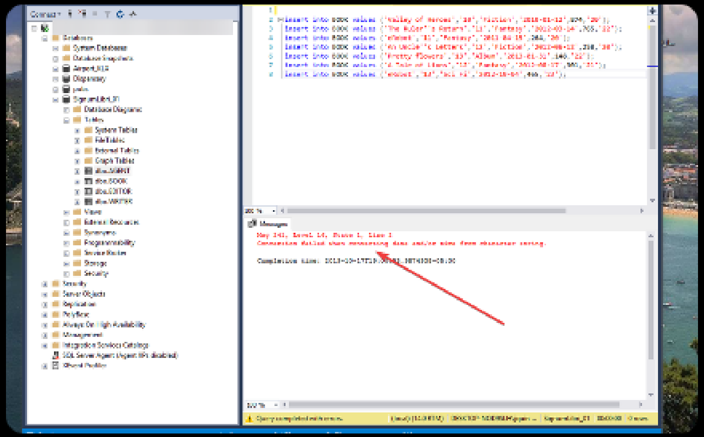
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```

```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```

```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
```

```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
```

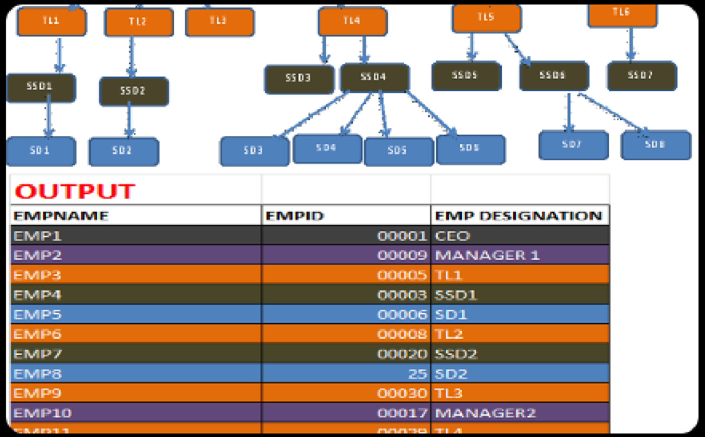
```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
```



```
insert into BOOK values ('Valley of Heroes','10','Fiction','2010-01-12',874,'20');
insert into BOOK values ('The Ruler''s Return','11','Fantasy','2012-03-14',765,'22');
insert into BOOK values ('[illegible]','11','Fantasy','2011-04-15',264,'20');
insert into BOOK values ('An Uncle''s Letters','12','Fiction','2012-06-12',258,'30');
insert into BOOK values ('Pretty Flowers','13','Album','2013-01-31',140,'22');
insert into BOOK values ('A Tale of Lions','12','Fantasy','2012-08-17',800,'32');
insert into BOOK values ('[illegible]','14','Sci Fi','2012-10-04',466,'32');
```

```
Msg 241, Level 16, State 1, Line 2
Conversion failed when converting date and/or time from character string.

Completion time: [illegible]
```



OUTPUT

EMPNAME	EMPID	EMP DESIGNATION
EMP1	00001	CEO
EMP2	00009	MANAGER 1
EMP3	00005	TL1
EMP4	00003	SSD1
EMP5	00006	SD1
EMP6	00008	TL2
EMP7	00020	SSD2
EMP8	25	SD2
EMP9	00030	TL3
EMP10	00017	MANAGER2
EMP11	[illegible]	TL4

Structured Language

SQL syntax follows a specific structure, including keywords, expressions, and clauses. Understanding the basic syntax is crucial for writing effective queries and commands.

Key Components

Learning the fundamental components of SQL syntax, such as such as SELECT, FROM, and WHERE, WHERE, lays the foundation for building more complex and powerful database operations.

Logical Operators

SQL syntax incorporates logical logical operators like AND, OR, and and NOT, which are essential for for defining conditions and filtering filtering data effectively.

Key points of SQL syntax

1

Statements

SQL syntax includes various statements such as SELECT, INSERT, UPDATE, and DELETE to perform specific tasks on the database.

2

Functions

SQL syntax incorporates a range of functions for data manipulation, transformation, and computation, providing flexibility in query construction. construction.

3

Clarity and Readability

Basic SQL syntax is designed for intuitive readability, making the code easy to easy to understand and maintain for database developers.

STATEMENTS USED IN SQL

SELECT Statement:

- Retrieve Data from a database.

SYNTAX:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

INSERT INTO Statement:

- Add a new record to the table.

SYNTAX:

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

UPDATE Statement:

- Modifying existing record in the table.

SYNTAX:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

FILTERING DATA USED IN SQL

WHERE Clause:

- Filter records returned by ‘SELECT’,’UPDATE’,’DELETE’ Statement.

SYNTAX:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

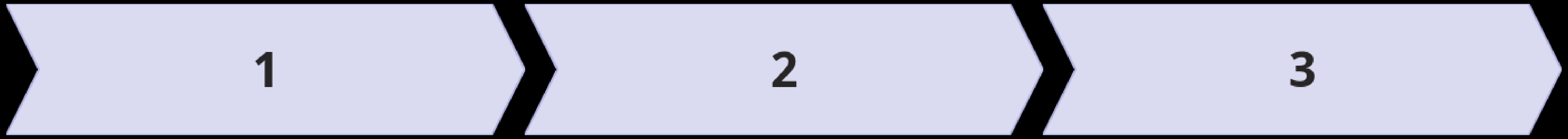
LOGICAL Operators:

- SQL provides logical operators such as AND, OR, and NOT to combine multiple conditions for more complex data filtering.

SYNTAX:

```
SELECT * FROM table_name WHERE condition1 AND condition2;
```

Joining tables to combine data



Inner Join

Combines rows from two or more tables based on a related related column between them them and excludes non-matching matching records.

Outer Join

Retrieves rows from one or more more tables, including all matching rows and displaying displaying NULL values for non-non-matching records.

Cross Join

Returns the Cartesian product of product of rows from multiple multiple tables, creating combinations of all rows between the tables.

REAL LIFE IMPLEMENTATION IN SQL

Digital Music Store Analysis :

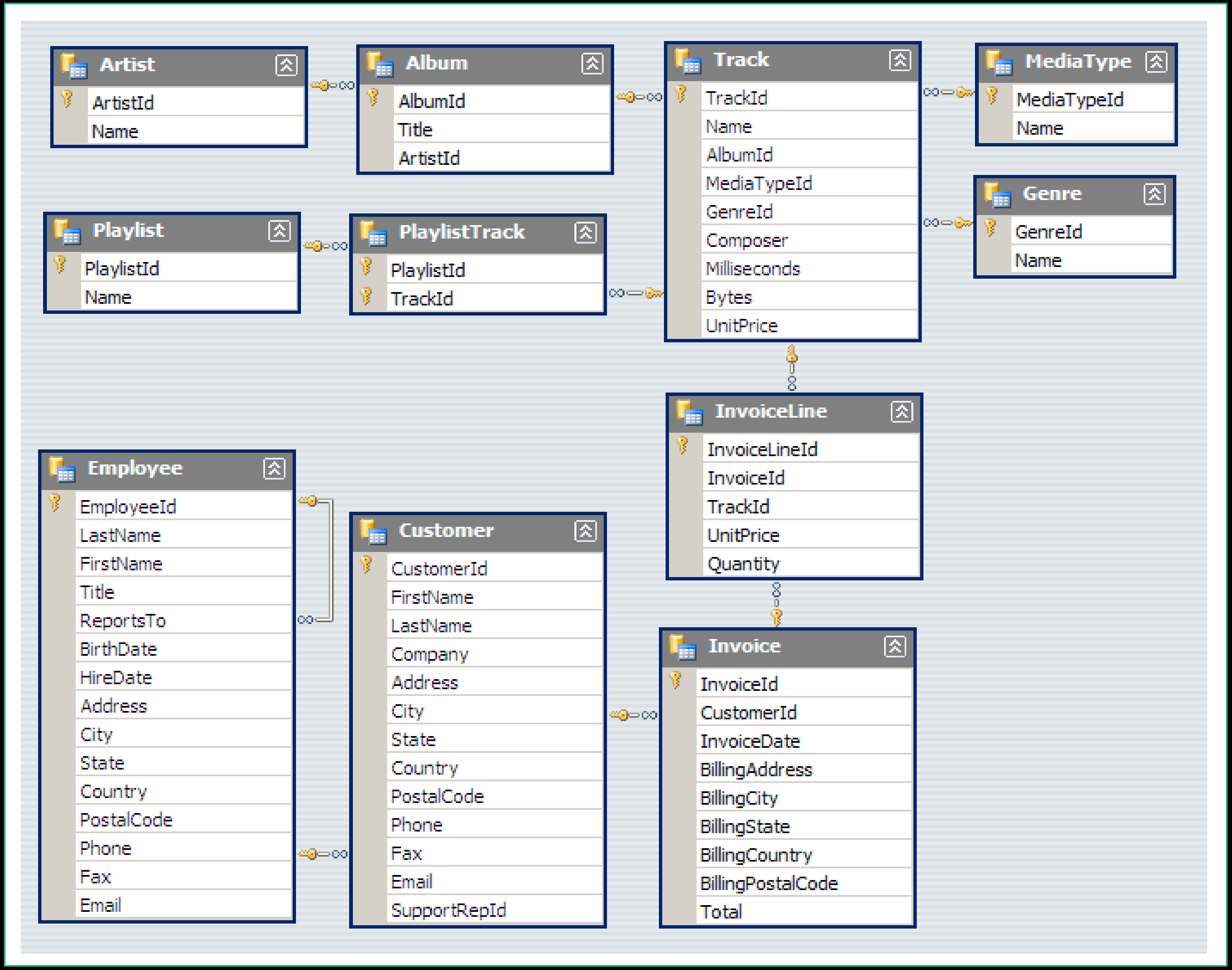
Analyzing a music store database involves extracting insights from the data it contains.

1. Understanding the Database Schema:

- Review the structure of the music store database. Understand the tables, relationships, and the data they store.

2. Connecting to the Database:

Use a SQL client or command-line interface to connect to the music store database.



3. Exploratory Queries:

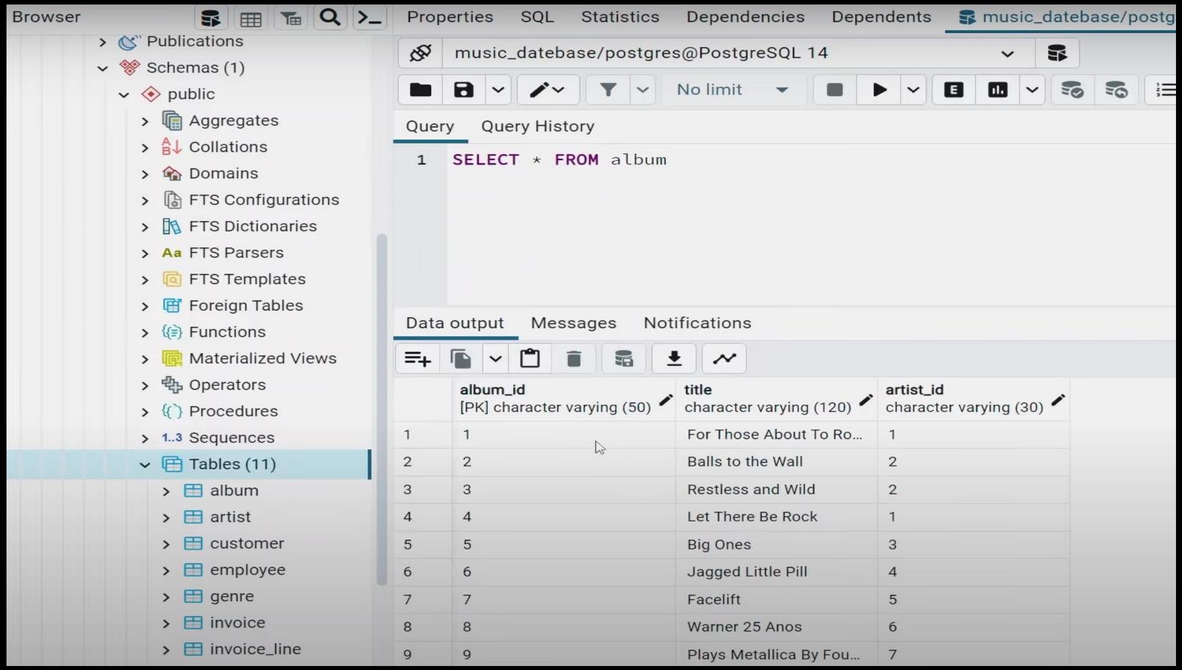
- Write basic queries to explore the data.

4. Data Cleaning and Transformation:

•Identify any inconsistencies or missing data.

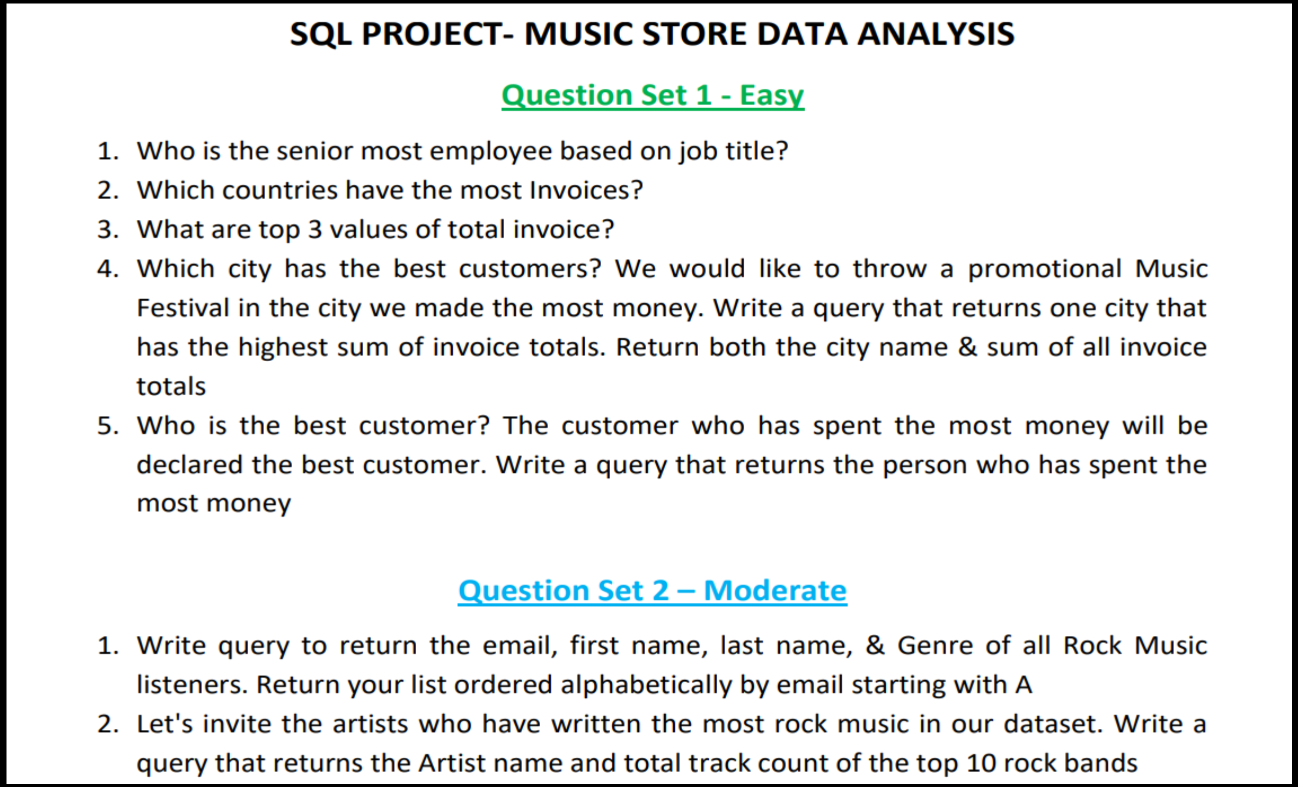
•Clean the data by removing duplicates, handling null values, or standardizing formats.

5. Performing Analysis:



```
SELECT * FROM album
```

	album_id [PK] character varying (50)	title character varying (120)	artist_id character varying (30)
1	1	For Those About To Ro...	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Fou...	7



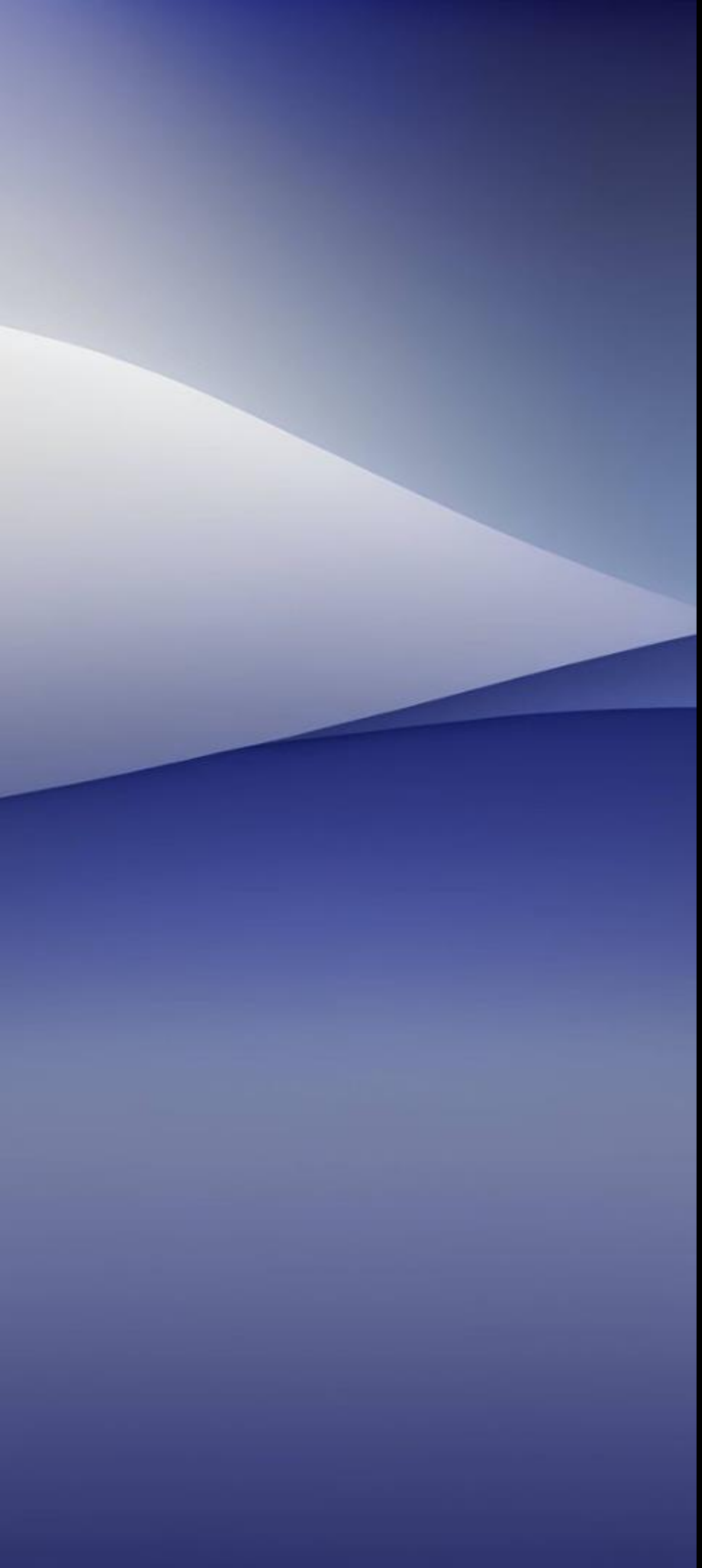
SQL PROJECT- MUSIC STORE DATA ANALYSIS

Question Set 1 - Easy

1. Who is the senior most employee based on job title?
2. Which countries have the most Invoices?
3. What are top 3 values of total invoice?
4. Which city has the best customers? We would like to throw a promotional Music Festival in the city we made the most money. Write a query that returns one city that has the highest sum of invoice totals. Return both the city name & sum of all invoice totals
5. Who is the best customer? The customer who has spent the most money will be declared the best customer. Write a query that returns the person who has spent the most money

Question Set 2 – Moderate

1. Write query to return the email, first name, last name, & Genre of all Rock Music listeners. Return your list ordered alphabetically by email starting with A
2. Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the Artist name and total track count of the top 10 rock bands



Real-life examples of SQL implementation

1

Data Analytics

SQL is widely used in data analytics for querying and processing large datasets to extract meaningful insights and trends.

2

Web Development

Many websites and web applications implement SQL for managing user data, content management, and dynamic content generation.

3

Business Reporting

SQL is integral to business reporting, providing the functionality to retrieve, analyze, and present data for decision-making purposes.

PROS:

1. **Ease of Use:** SQL has a relatively simple syntax, making it easy to learn and use, especially for querying and managing relational databases.

2. **Standardization:** Being a standardized language, SQL allows for portability across various database platforms, making it easier to transfer skills and code between systems.

3. **Robustness:** It can handle complex queries and database operations, supporting a wide range of functionalities like data retrieval, manipulation, and administration.

4. **Scalability:** SQL databases can scale vertically by adding more resources to a single server or horizontally by distributing data across multiple servers.

5. **Data Integrity:** SQL supports constraints (e.g., foreign keys, unique keys) ensuring data integrity, reducing the risk of inconsistent or incorrect data.

6. **Security:** SQL databases provide strong security features, including user access controls, encryption, and authentication mechanisms, safeguarding sensitive data.

CONS :

1. **Limited Support for Non-Relational Data**: SQL is primarily designed for relational databases, which might not be the ideal choice for handling non-relational, unstructured, or hierarchical data.

2. **Performance Bottlenecks:** Poorly optimized queries or database design can lead to performance issues, especially with large datasets or complex queries.

3. **Learning Curve:** While basic SQL is easy to learn, mastering advanced concepts, optimization techniques, and best practices might require significant time and experience.

4. **Vendor Lock-in:** Different database vendors might have proprietary extensions or implementations, potentially leading to vendor lock-in for advanced functionalities.

5. **Complex Joins:** Complex joins across multiple tables can be challenging to write, understand, and optimize, impacting query performance.

6. **Scaling Challenges:** While SQL databases can scale, horizontal scaling (across multiple servers) can be more complex compared to some NoSQL databases designed for this purpose.

TOOLS

SQL Query Development and Management:

1.SQL Server Management Studio (SSMS): Microsoft's tool for managing SQL Server databases, offering a rich environment for writing queries, managing databases, and performing administrative tasks.

2.MySQL Workbench: An official tool for MySQL, providing SQL development, database administration, and data modelling capabilities.

3.Oracle SQL Developer: Oracle's tool for developing and managing Oracle databases, featuring SQL development, query optimization, and database administration .

Data Visualization and Business Intelligence:

1.Tableau: A powerful BI tool that can connect to SQL databases for data visualization, dashboards, and analytics.

2.Power BI: Microsoft's business analytics tool supporting SQL databases, allowing data visualization, reporting, and dashboard creation.

3.Looker: A data exploration and visualization platform that can connect to SQL databases for analyzing and visualizing data.

Conclusion

SQL Proficiency	Deepening knowledge through practical application and continuous learning continuous learning
Advanced Concepts	Explore advanced SQL features such as stored procedures, triggers, and triggers, and optimization techniques
Career Opportunities	Utilize SQL skills to pursue opportunities in data analytics administration, administration, business intelligence, and database

REFERENCES

1) "SQL Performance Explained" by Markus Winand

2) "Learning SQL" by Alan Beaulieu

3) "SQL Antipatterns: Avoiding the Pitfalls of Database Programming" by Bill Karwin

4) Research paper – “transaction Management in SQL Server: Authors” by Microsoft Corporation

5) “Comparative Study of NoSQL Databases” :Authors: Shatakshi Asthana, Dr. Manish Prateek

6) Sample project GitHub: link:https://github.com/ankitsin1/Music_Store_Analysis_SQL

THANK YOU