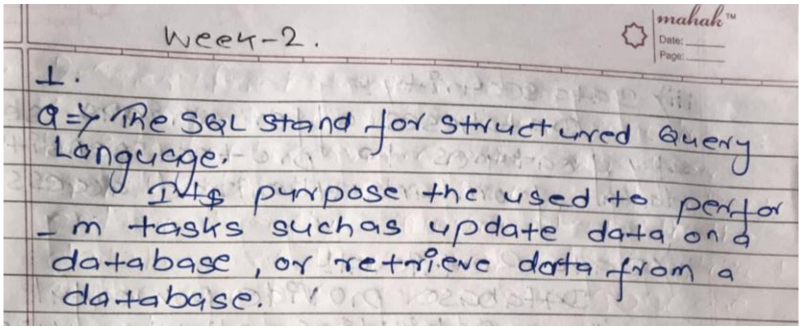
Week 2: Introduction to SQL

1. Define Terms:

I. What does SQL stand for, and what is its purpose?



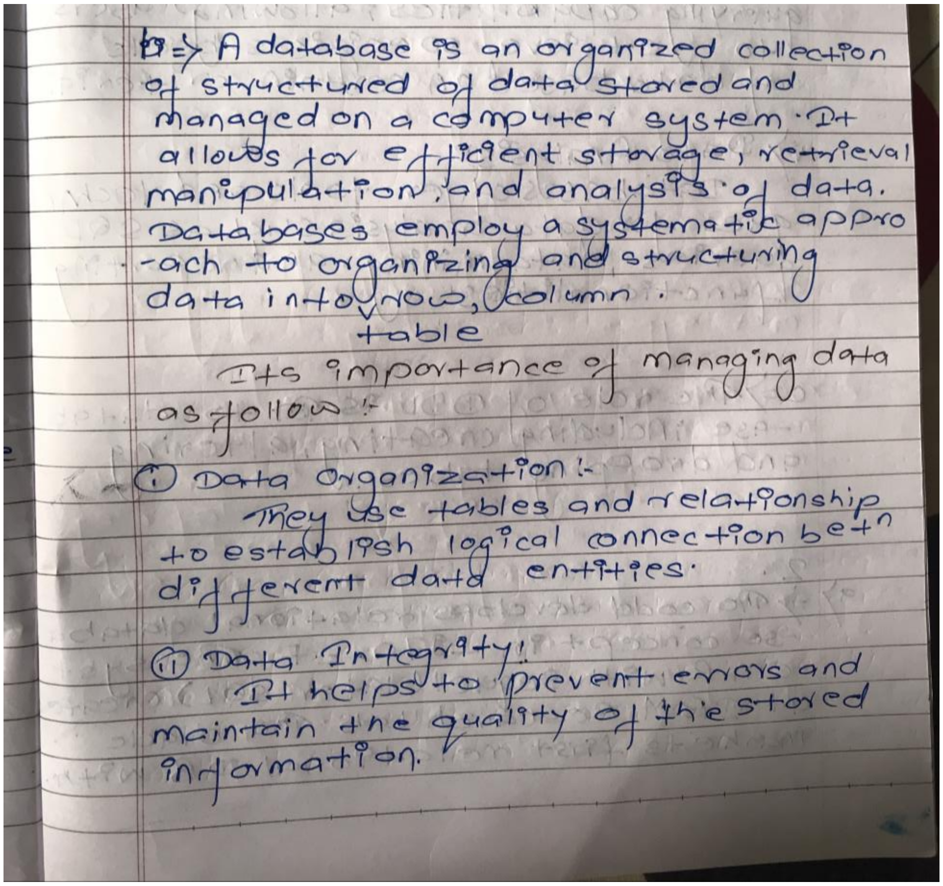
Week-2.

1.

a=> The SQL stand for structured Query Language.

Its purpose the used to perform tasks suchas update data on a database, or retrieve data from a database.

II. Explain the concept of a database and its importance of managing data.



1) => A database is an organized collection of structured of data stored and managed on a computer system. It allows for efficient storage, retrieval manipulation, and analysis of data. Databases employ a systematic appro-ach to organizing and structuring data into table row, column.

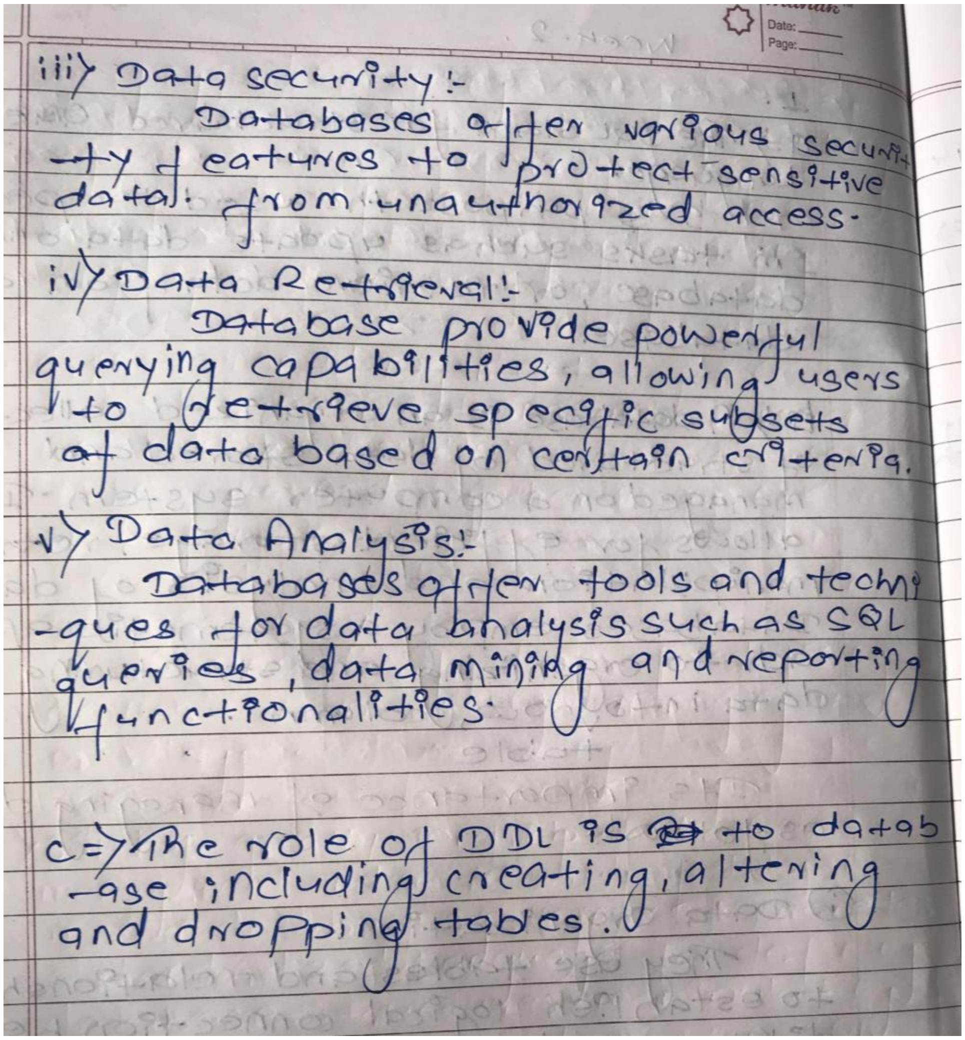
Its importance of managing data as follow :-

(i) Data Organization :-

They use tables and relationship to establish logical connection bet^{n} different data entities.

(ii) Data Integrity :-

It helps to prevent errors and maintain the quality of the stored information.



Date:
Page:

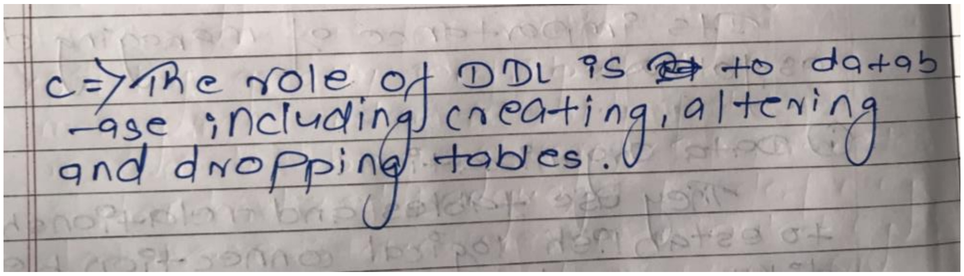
iii) Data security:-
Databases offer various security features to protect sensitive data: from unauthorized access.

iv) Data Retrieval:-
Database provide powerful querying capabilities, allowing users to retrieve specific subsets of data based on certain criteria.

v) Data Analysis:-
Databases offer tools and techniques for data analysis such as SQL queries, data mining and reporting functionalities.

c=> The role of DDL is to database including creating, altering and dropping tables.

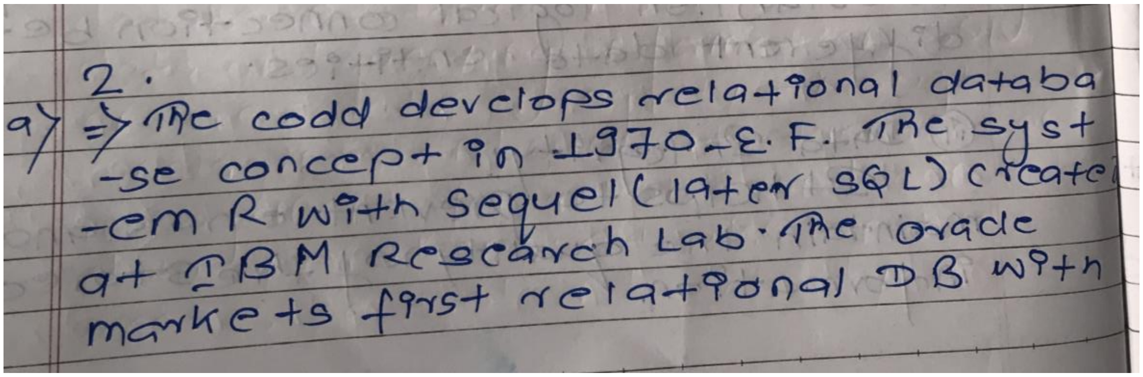
III. What is the role of Data Definition Language (DDL) in SQL?



c=> The role of DDL is to database including creating, altering and dropping tables.

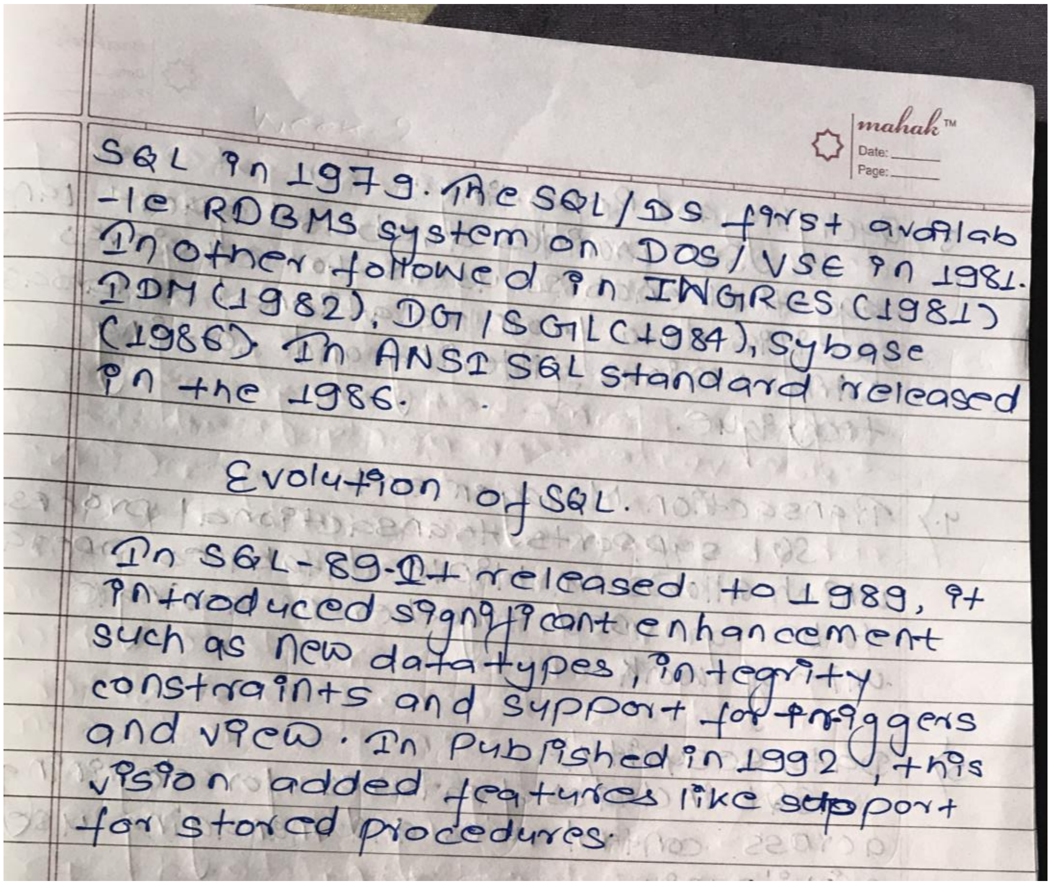
2. History and Role of SQL:

i. Describe the history and evolution of SQL.



2.

a) => The codd develops relational database concept in 1970 - E.F. The system R with sequel (later SQL) created at IBM Research Lab. The oracle markets first relational DB with

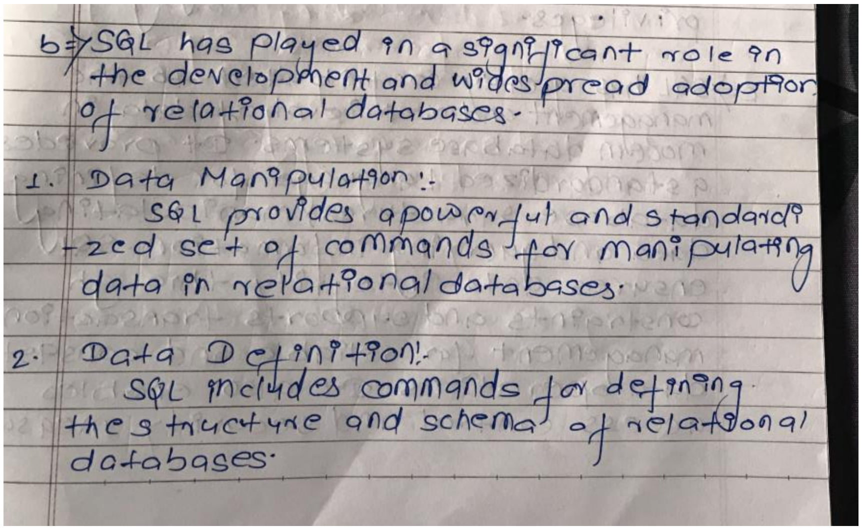


SQL in 1979. The SQL/DS first available RDBMS system on DOS/VSE in 1981. In other followed in INGRES (1981) IDM (1982), DG/SGL (1984), Sybase (1986). In ANSI SQL standard released in the 1986.

Evolution of SQL.

In SGL-89 - It released to 1989, it introduced significant enhancement such as new datatypes, integrity constraints and support for triggers and view. In Published in 1992, this vision added features like support for stored procedures.

ii. Discuss the significance of SQL in the development of relational databases.

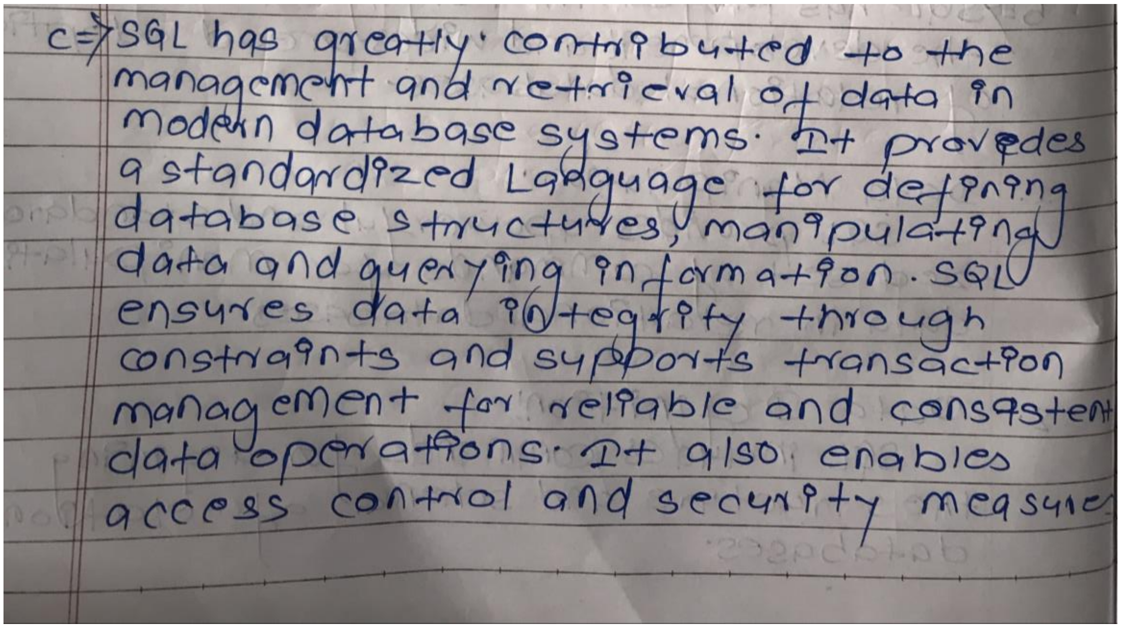


b=>SQL has played in a significant role in the development and widespread adoption of relational databases.

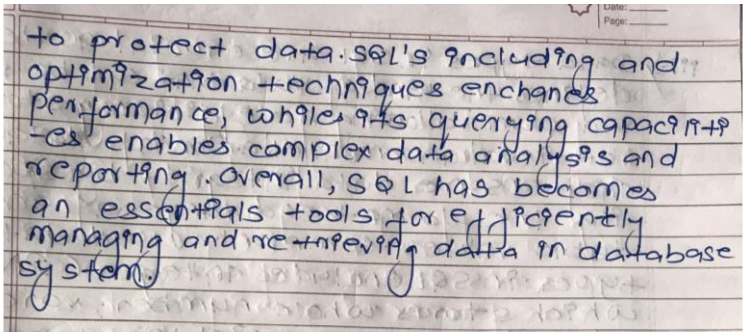
1. Data Manipulation:-
SQL provides apowerful and standardized set of commands for manipulating data in relational databases.

2. Data Definition:-
SQL includes commands for defining the structure and schema of relational databases.

iii. Explain how SQl has contributed to the management and retrieval of data in modern database systems.



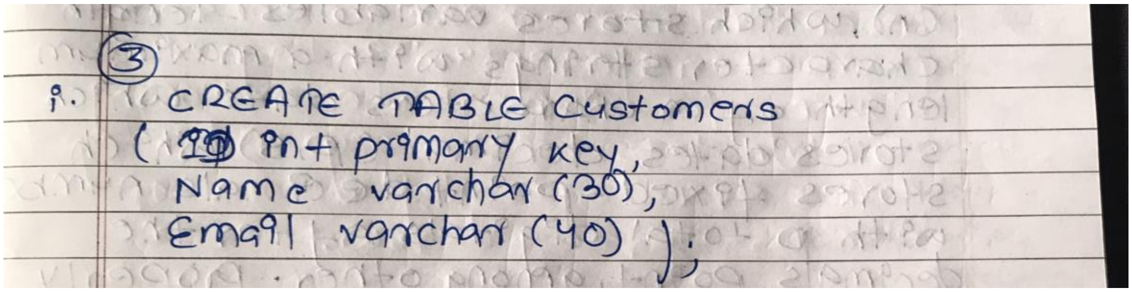
c=>SQL has greatly contributed to the management and retrieval of data in modern database systems. It provides a standardized Language for defining database structures, manipulating data and querying information. SQL ensures data integrity through constraints and supports transaction management for reliable and consistent data operations. It also enables access control and security measures



to protect data. SQL's including and optimization techniques enchanes performance, while its querying capacities enables complex data analysis and reporting. Overall, SQL has becomes an essentials tools for efficiently managing and retrieving data in database system.

3. Define a Database using SQL Data Definition Language:

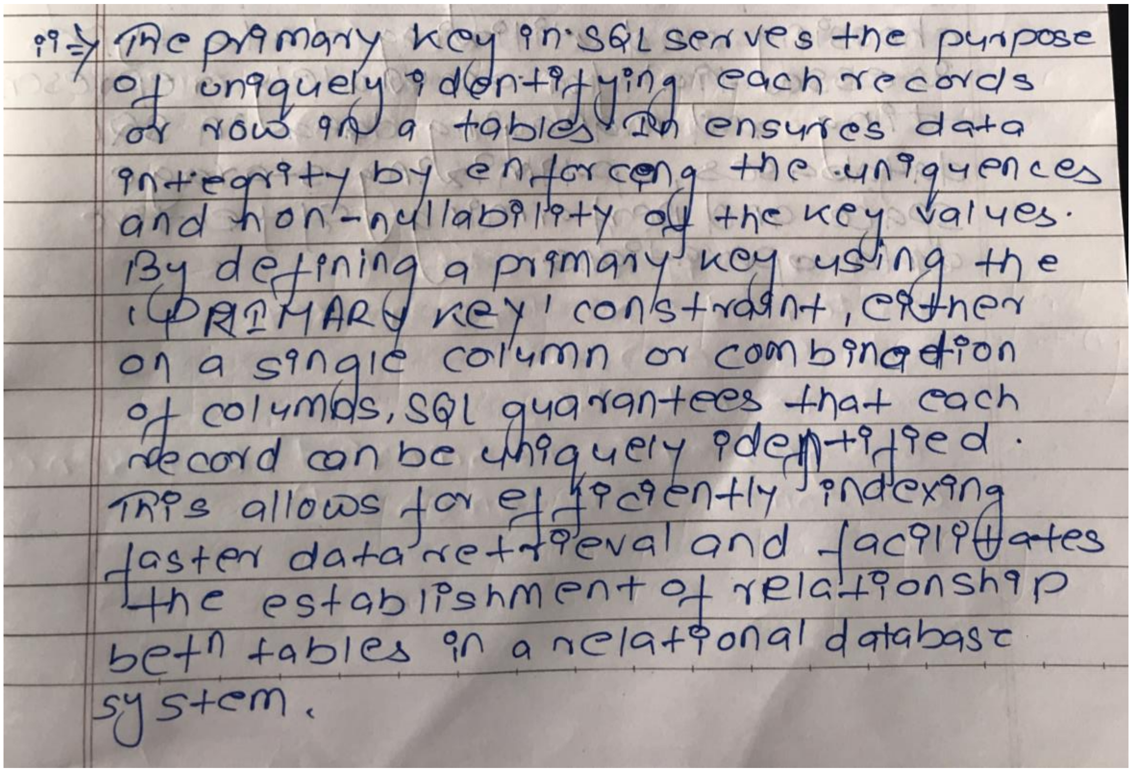
i. Write a SQL statement to create a new table called “Customers” with columns for “ID”, “Name”, and “Email”.



③

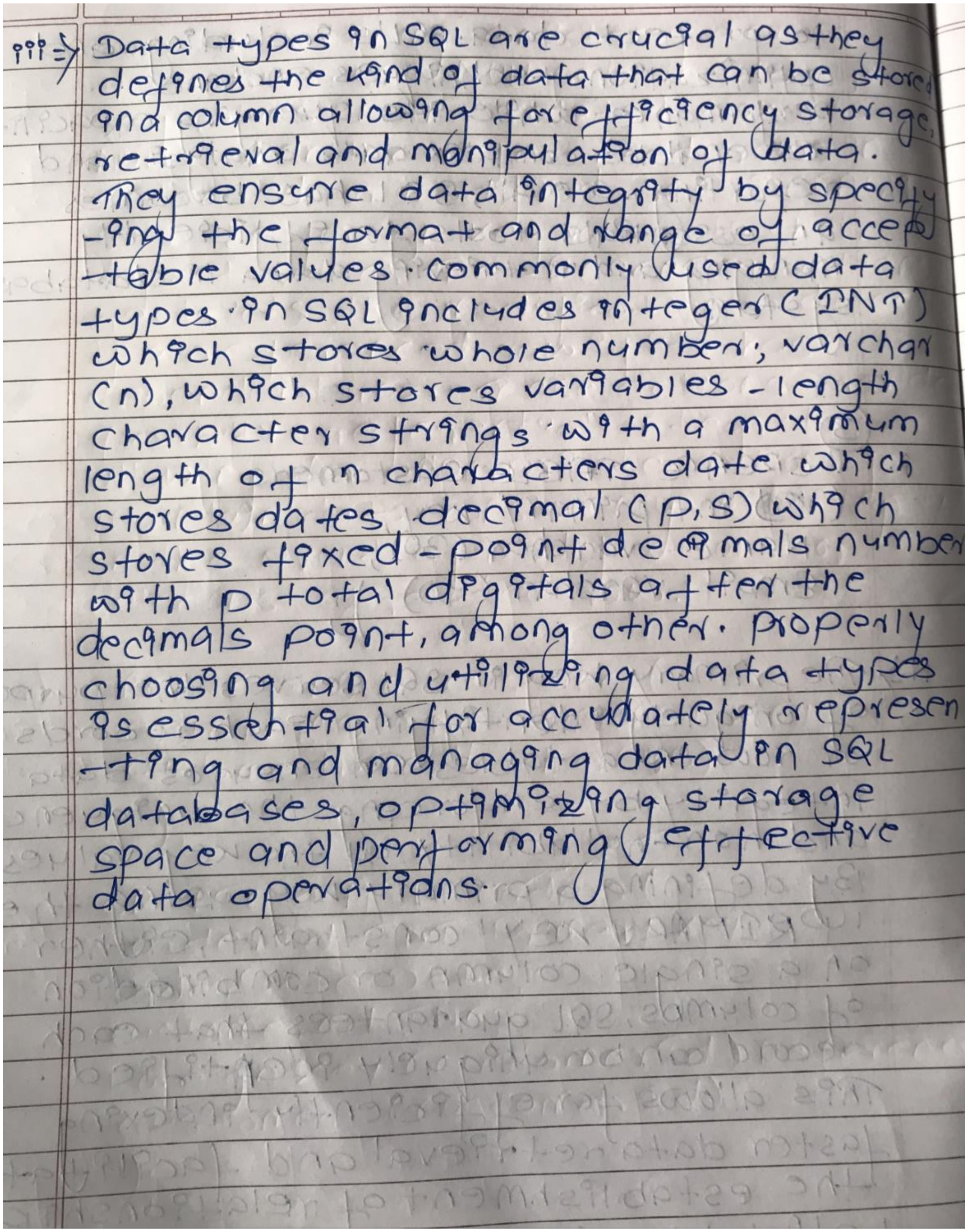
i. CREATE TABLE Customers
(ID int primary key,
Name varchar (30),
Email varchar (40));

ii. Explain the purpose of primary keys and how they are defined in SQL.



ii ⇒ The primary key in SQL serves the purpose of uniquely identifying each records or row in a tables. It ensures data integrity by enforcing the uniquences and non-nullability of the key values. By defining a primary key using the 'PRIMARY key' constraint, either on a single column or combination of columns, SQL guarantees that each record can be uniquely identified. This allows for efficiently indexing faster data retrieval and facilitates the establishment of relationship bet^{n} tables in a relational database system.

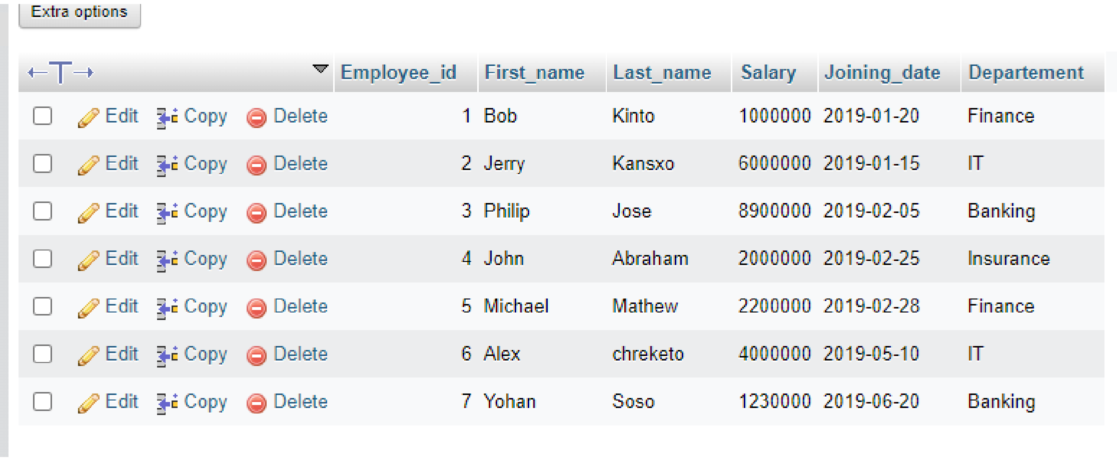
iii. Discuss the importance of data types in SQL and provide examples of commonly used data types.



iii ⇒ Data types in SQL are crucial as they defines the kind of data that can be stored and column allowing for efficiency storage, retrieval and manipulation of data. They ensure data integrity by specifying the format and range of acceptable values. commonly used data types in SQL includes integer (INT) which stores whole number; varchar (n), which stores variables - length character strings with a maximum length of n characters date which stores dates, decimal (p,s) which stores fixed - point decimals number with p total digitals after the decimals point, among other. properly choosing and utilizing data types is essential for accurately representing and managing data in SQL databases, optimizing storage space and performing effective data operations.

4. Write Single Table Queries Using SQL:

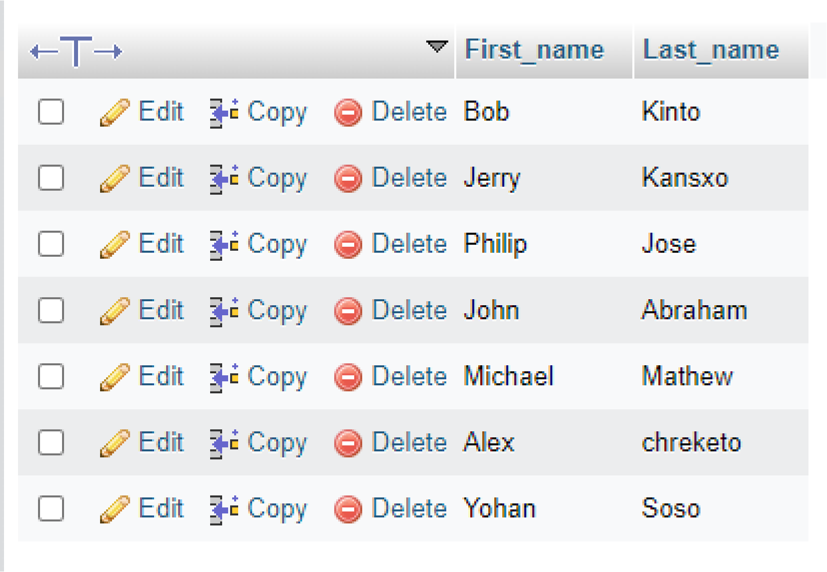
- Get all employees



Extra options

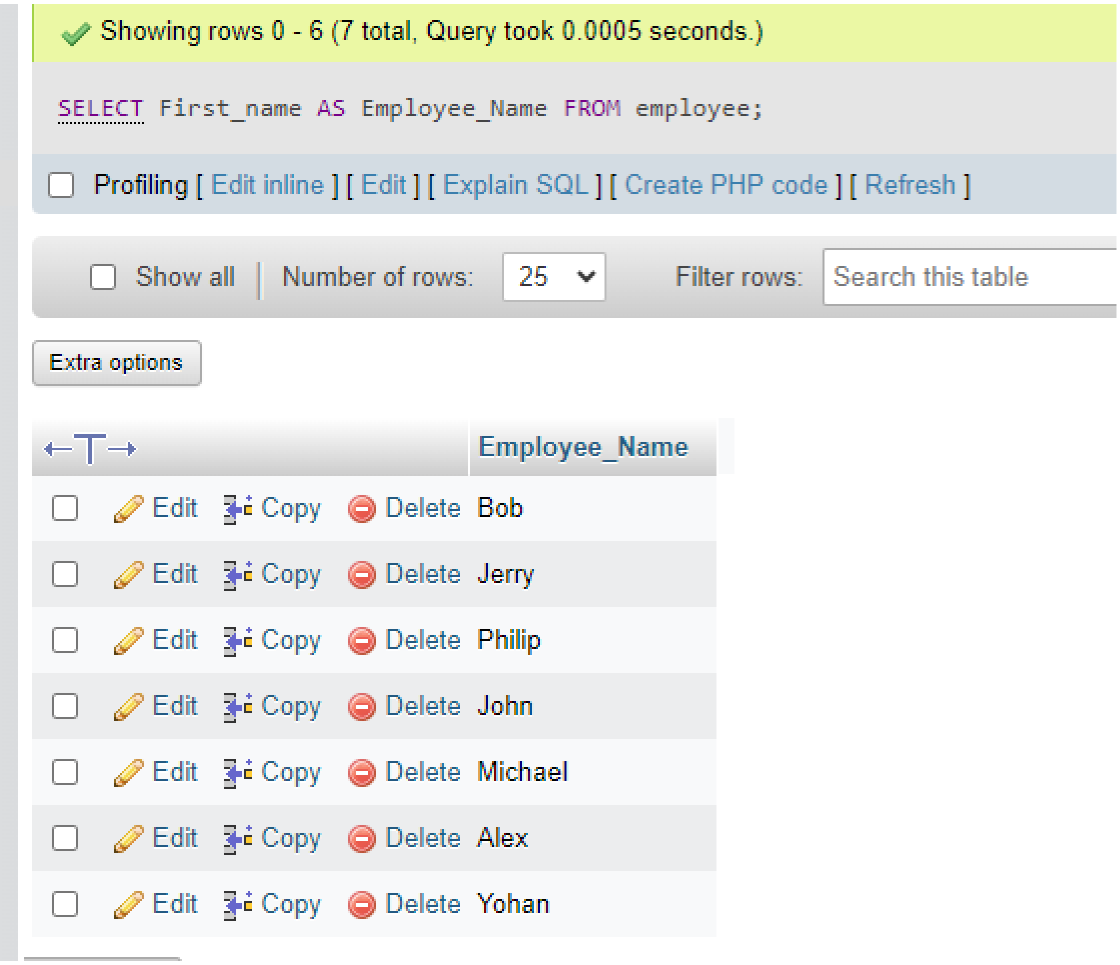
←T→				Employee_id	First_name	Last_name	Salary	Joining_date	Departement
□	Edit	Copy	Delete	1	Bob	Kinto	1000000	2019-01-20	Finance
□	Edit	Copy	Delete	2	Jerry	Kansxo	6000000	2019-01-15	IT
□	Edit	Copy	Delete	3	Philip	Jose	8900000	2019-02-05	Banking
□	Edit	Copy	Delete	4	John	Abraham	2000000	2019-02-25	Insurance
□	Edit	Copy	Delete	5	Michael	Mathew	2200000	2019-02-28	Finance
□	Edit	Copy	Delete	6	Alex	chreketo	4000000	2019-05-10	IT
□	Edit	Copy	Delete	7	Yohan	Soso	1230000	2019-06-20	Banking

- Display the first name and last name of all employees.



←T→				First_name	Last_name
□	Edit	Copy	Delete	Bob	Kinto
□	Edit	Copy	Delete	Jerry	Kansxo
□	Edit	Copy	Delete	Philip	Jose
□	Edit	Copy	Delete	John	Abraham
□	Edit	Copy	Delete	Michael	Mathew
□	Edit	Copy	Delete	Alex	chreketo
□	Edit	Copy	Delete	Yohan	Soso

- Display all the values of the “First_Name” column using the alias “Employee Name”



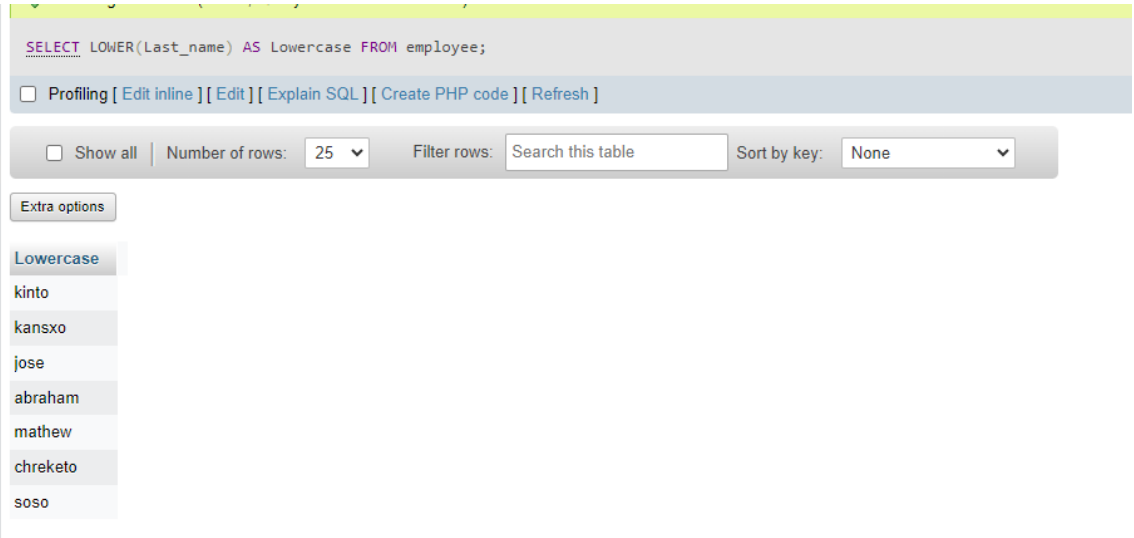
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Extra options

✓ Showing rows 0 - 6 (7 total, Query took 0.0005 seconds.)

```
SELECT First_name AS Employee_Name FROM employee;
```

←T→				Employee_Name
☐	Edit	Copy	Delete	Bob
☐	Edit	Copy	Delete	Jerry
☐	Edit	Copy	Delete	Philip
☐	Edit	Copy	Delete	John
☐	Edit	Copy	Delete	Michael
☐	Edit	Copy	Delete	Alex
☐	Edit	Copy	Delete	Yohan

- Get all “Last_Name” in lowercase.

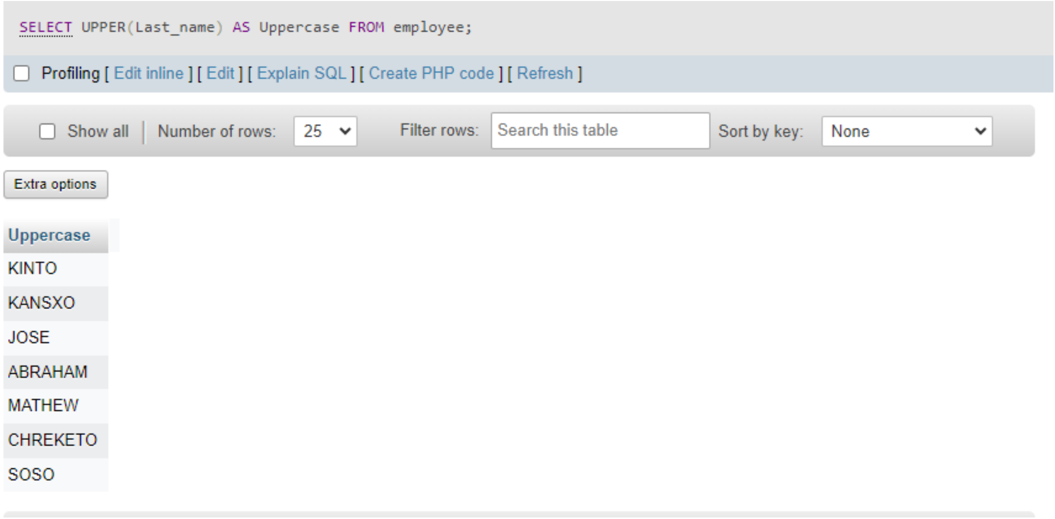


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options

```
SELECT LOWER(Last_name) AS Lowercase FROM employee;
```

Lowercase
kinto
kansxo
jose
abraham
mathew
chreketo
soso

- Get all “Last_Name” in uppercase.

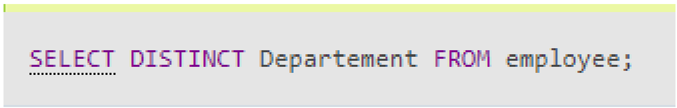


```
SELECT UPPER(Last_name) AS Uppercase FROM employee;
```

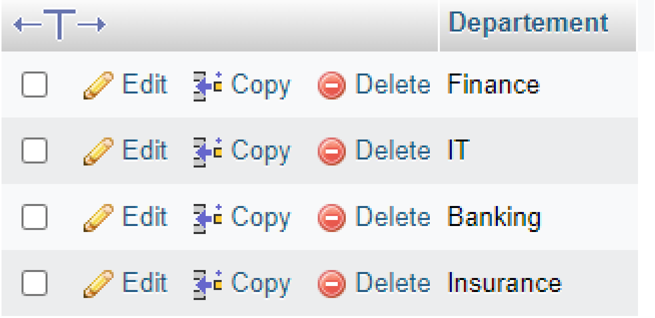
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

Uppercase
KINTO
KANSXO
JOSE
ABRAHAM
MATHEW
CHREKETO
SOSO

- Get unique “DEPARTMENT”.

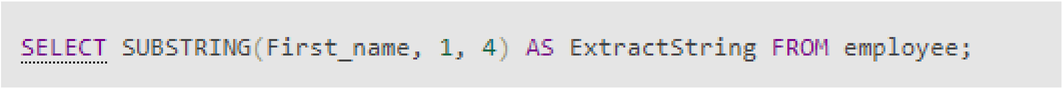


```
SELECT DISTINCT Departement FROM employee;
```

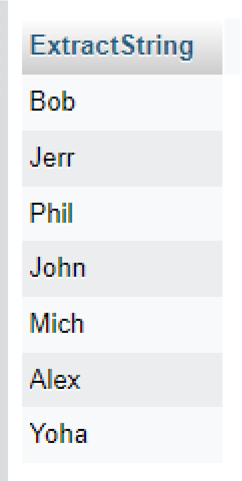


←T→	Departement
□ Edit Copy Delete	Finance
□ Edit Copy Delete	IT
□ Edit Copy Delete	Banking
□ Edit Copy Delete	Insurance

- Get the first 4 characters of “FIRST_NAME” column.

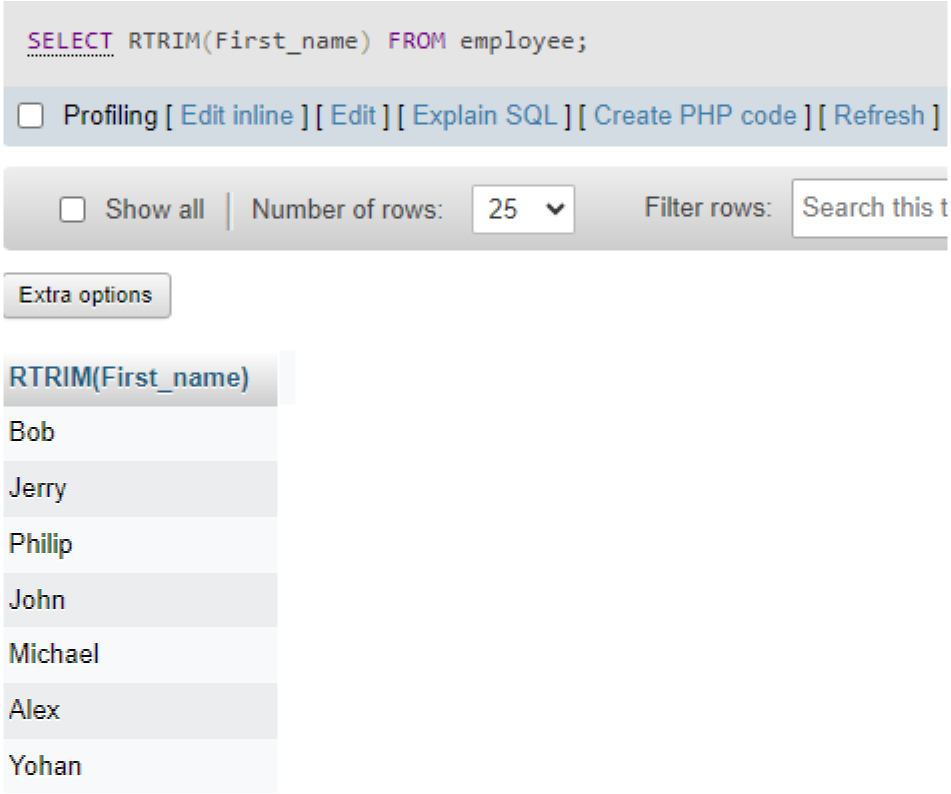


```
SELECT SUBSTRING(First_name, 1, 4) AS ExtractString FROM employee;
```



ExtractString
Bob
Jerr
Phil
John
Mich
Alex
Yoha

- Get all values from the “FIRST_NAME” column after removing white space on the right.

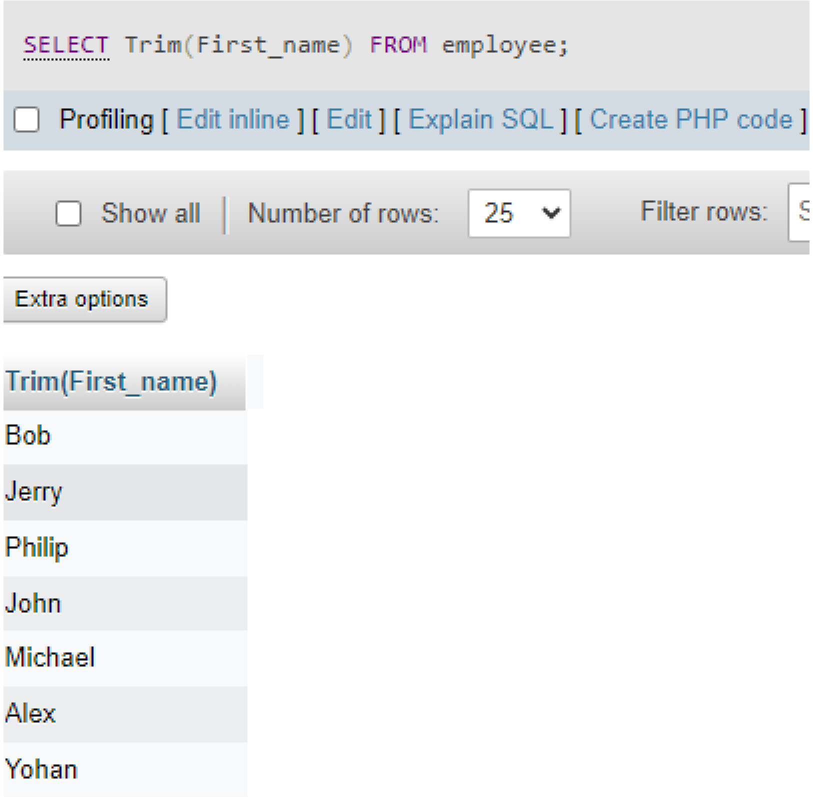


```
SELECT RTRIM(First_name) FROM employee;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
☐ Show all
Number of rows: 25
Filter rows: Search this t
Extra options

RTRIM(First_name)
Bob
Jerry
Philip
John
Michael
Alex
Yohan

- Get all values from the “FIRST_NAME” column after removing white space on the left.



```
SELECT Trim(First_name) FROM employee;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]
□ Show all | Number of rows: 25
Filter rows:
Extra options

Trim(First_name)
Bob
Jerry
Philip
John
Michael
Alex
Yohan

- Write the syntax to create the "employee" table.

Ans:

Create Table employee(

Employee_id int(30) ,

First_name varchar(50) ,

Last_name varchar(50) ,

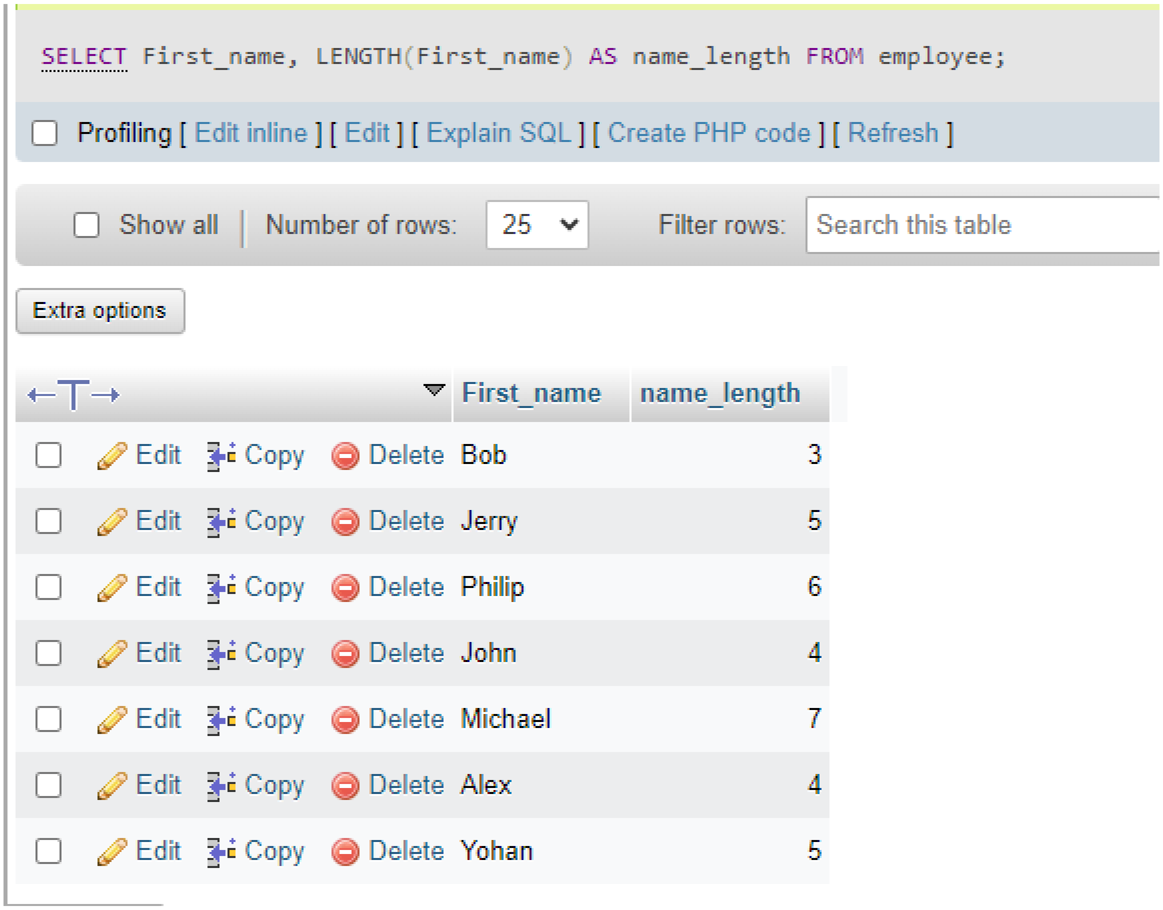
Salary int(40) ,

Joining_date date ,

Department varchar(50)

);

- Get the length of the text in the "First_name" column.



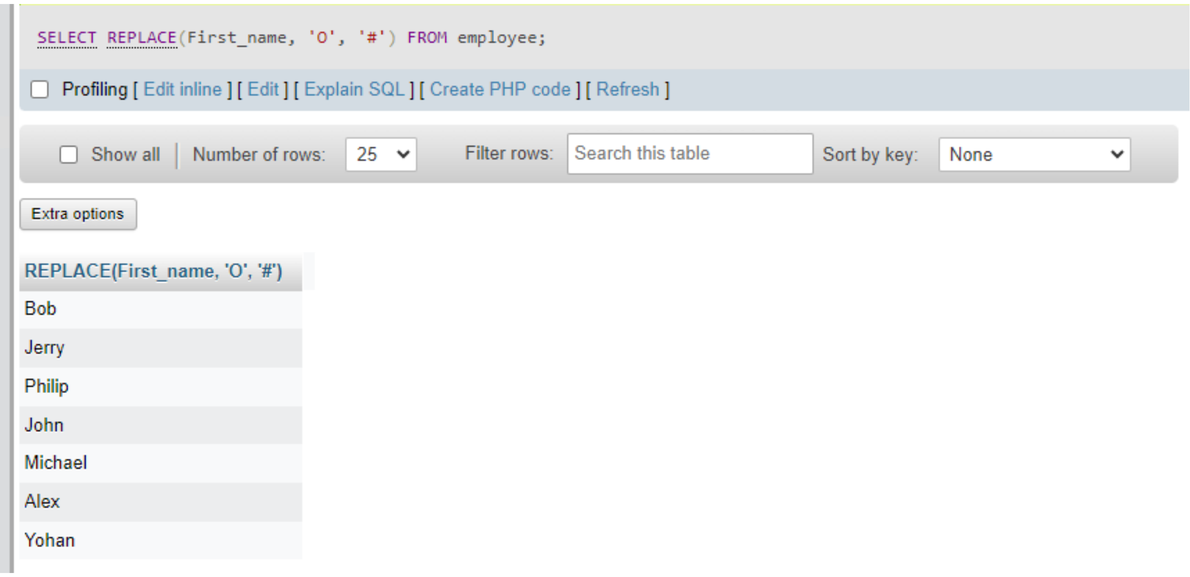
```
SELECT First_name, LENGTH(First_name) AS name_length FROM employee;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options
Edit
Copy
Delete

Number of rows: 25

First_name	name_length
Bob	3
Jerry	5
Philip	6
John	4
Michael	7
Alex	4
Yohan	5

- Get the employee’s first name after replacing ‘o’ with ‘#’.



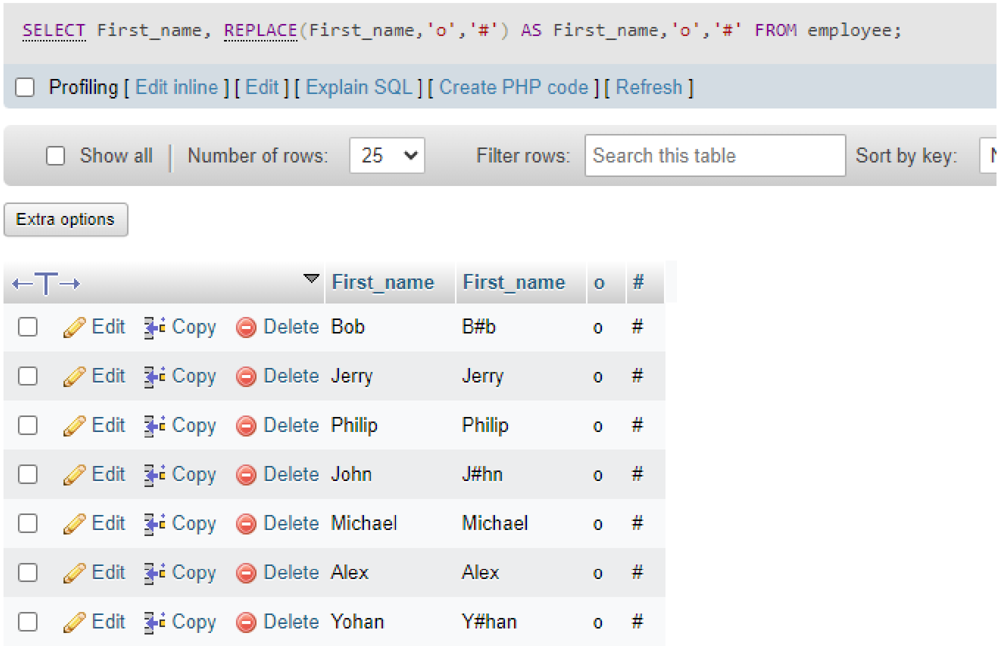
```
SELECT REPLACE(First_name, 'O', '#') FROM employee;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options

Number of rows: 25

Sort by key: None

REPLACE(First_name, 'O', '#')
Bob
Jerry
Philip
John
Michael
Alex
Yohan



```
SELECT First_name, REPLACE(First_name,'o','#') AS First_name,'o','#' FROM employee;
```

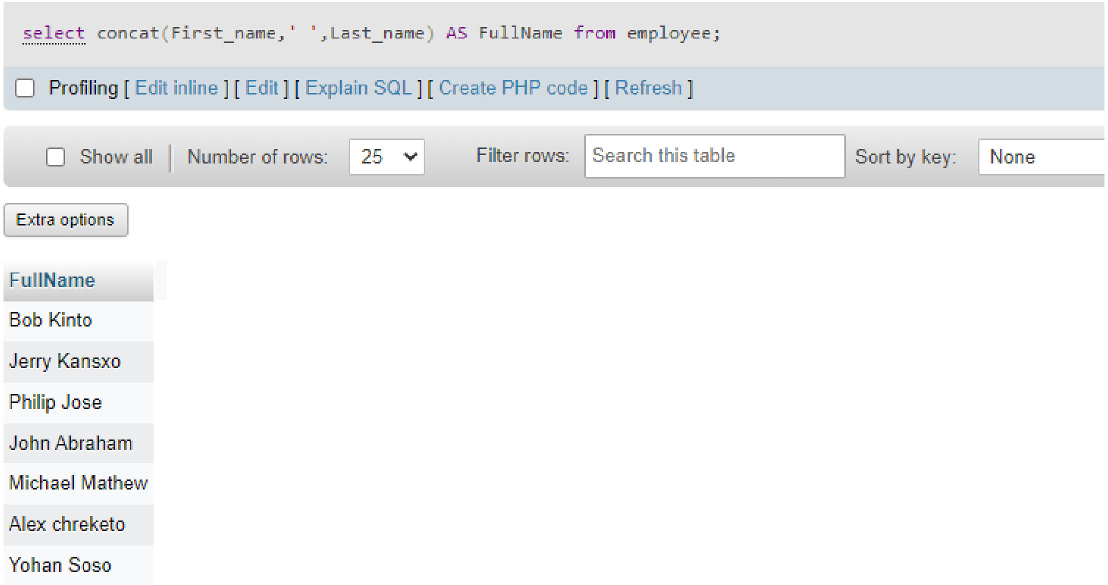
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key:

Extra options

	First_name	First_name	o	#
□ Edit Copy Delete	Bob	B#b	o	#
□ Edit Copy Delete	Jerry	Jerry	o	#
□ Edit Copy Delete	Philip	Philip	o	#
□ Edit Copy Delete	John	J#hn	o	#
□ Edit Copy Delete	Michael	Michael	o	#
□ Edit Copy Delete	Alex	Alex	o	#
□ Edit Copy Delete	Yohan	Y#han	o	#

- Get the employee's last name and first name in a single column separated by a '_'.



```
select concat(First_name,' ',Last_name) AS FullName from employee;
```

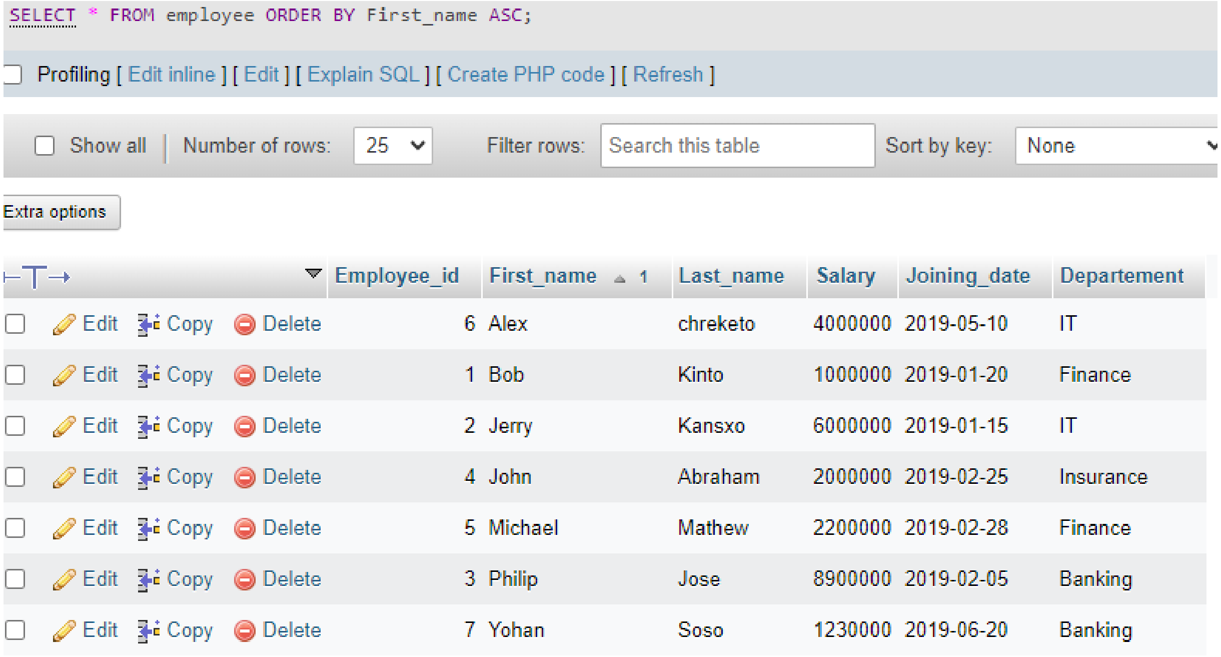
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

FullName
Bob Kinto
Jerry Kansxo
Philip Jose
John Abraham
Michael Mathew
Alex chreketo
Yohan Soso

- Get all employees in ascending order by first name.



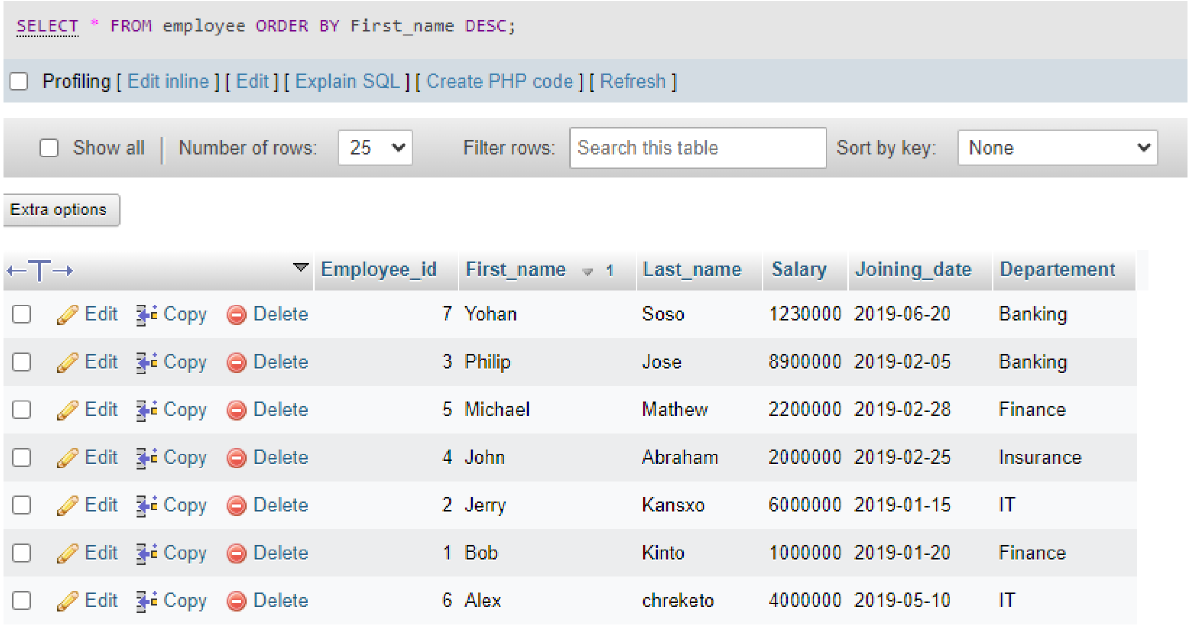
```
SELECT * FROM employee ORDER BY First_name ASC;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Extra options

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	Employee_id	First_name ▲ 1	Last_name	Salary	Joining_date	Departement
□ Edit Copy Delete	6	Alex	chreketo	4000000	2019-05-10	IT
□ Edit Copy Delete	1	Bob	Kinto	1000000	2019-01-20	Finance
□ Edit Copy Delete	2	Jerry	Kansxo	6000000	2019-01-15	IT
□ Edit Copy Delete	4	John	Abraham	2000000	2019-02-25	Insurance
□ Edit Copy Delete	5	Michael	Mathew	2200000	2019-02-28	Finance
□ Edit Copy Delete	3	Philip	Jose	8900000	2019-02-05	Banking
□ Edit Copy Delete	7	Yohan	Soso	1230000	2019-06-20	Banking

- Get all employees in descending order by first name.



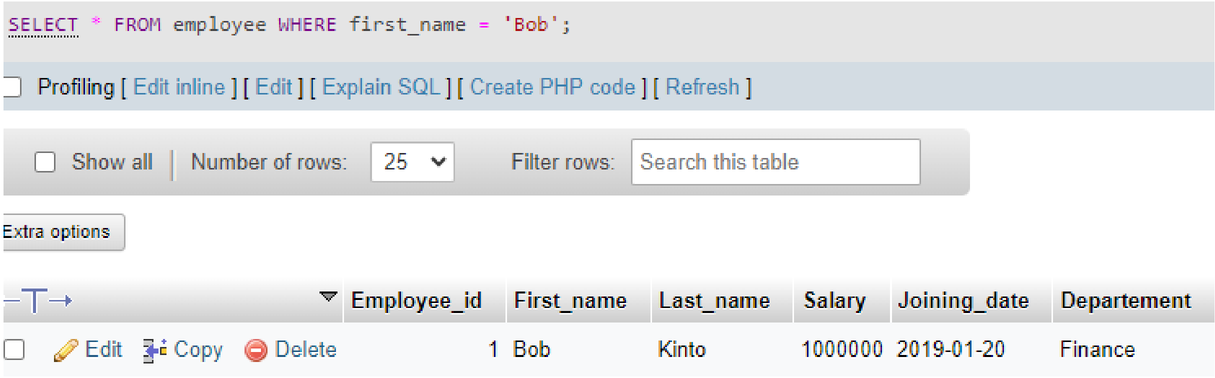
```
SELECT * FROM employee ORDER BY First_name DESC;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Extra options

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	Employee_id	First_name ▼ 1	Last_name	Salary	Joining_date	Departement
□ Edit Copy Delete	7	Yohan	Soso	1230000	2019-06-20	Banking
□ Edit Copy Delete	3	Philip	Jose	8900000	2019-02-05	Banking
□ Edit Copy Delete	5	Michael	Mathew	2200000	2019-02-28	Finance
□ Edit Copy Delete	4	John	Abraham	2000000	2019-02-25	Insurance
□ Edit Copy Delete	2	Jerry	Kansxo	6000000	2019-01-15	IT
□ Edit Copy Delete	1	Bob	Kinto	1000000	2019-01-20	Finance
□ Edit Copy Delete	6	Alex	chreketo	4000000	2019-05-10	IT

- Get employees whose first name is “Bob”.



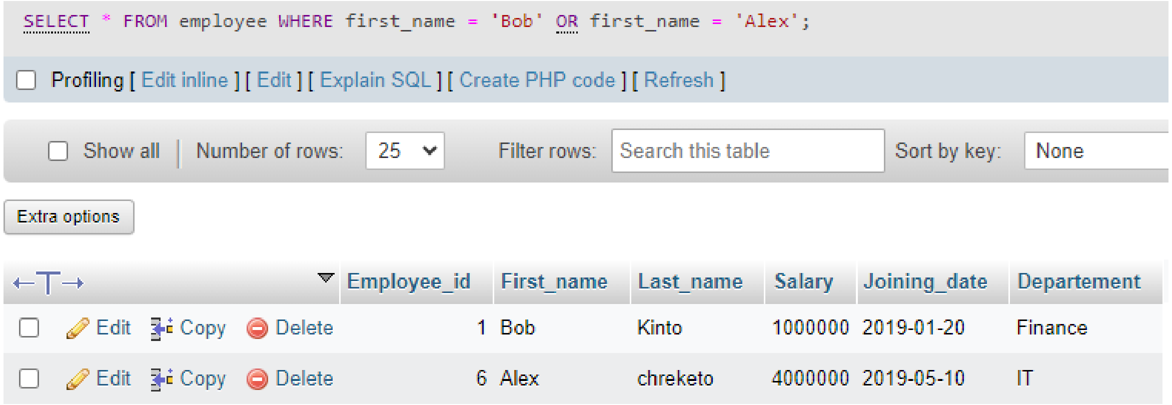
```
SELECT * FROM employee WHERE first_name = 'Bob';
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options

Number of rows: 25

	Employee_id	First_name	Last_name	Salary	Joining_date	Departement
□ Edit Copy Delete	1	Bob	Kinto	1000000	2019-01-20	Finance

- Get employees whose first name is “Bob” or “Alex”.



```
SELECT * FROM employee WHERE first_name = 'Bob' OR first_name = 'Alex';
```

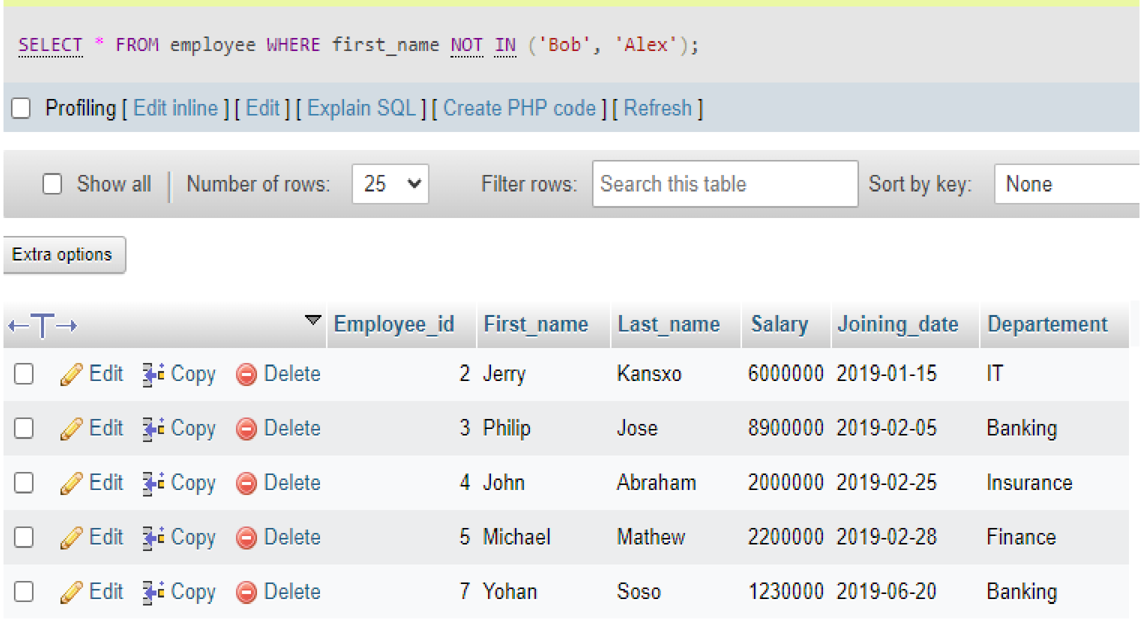
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options

Number of rows: 25

Sort by key: None

	Employee_id	First_name	Last_name	Salary	Joining_date	Departement
□ Edit Copy Delete	1	Bob	Kinto	1000000	2019-01-20	Finance
□ Edit Copy Delete	6	Alex	chreketo	4000000	2019-05-10	IT

- Get employees whose first name is neither “Bob” nor “Alex”.



```
SELECT * FROM employee WHERE first_name NOT IN ('Bob', 'Alex');
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options

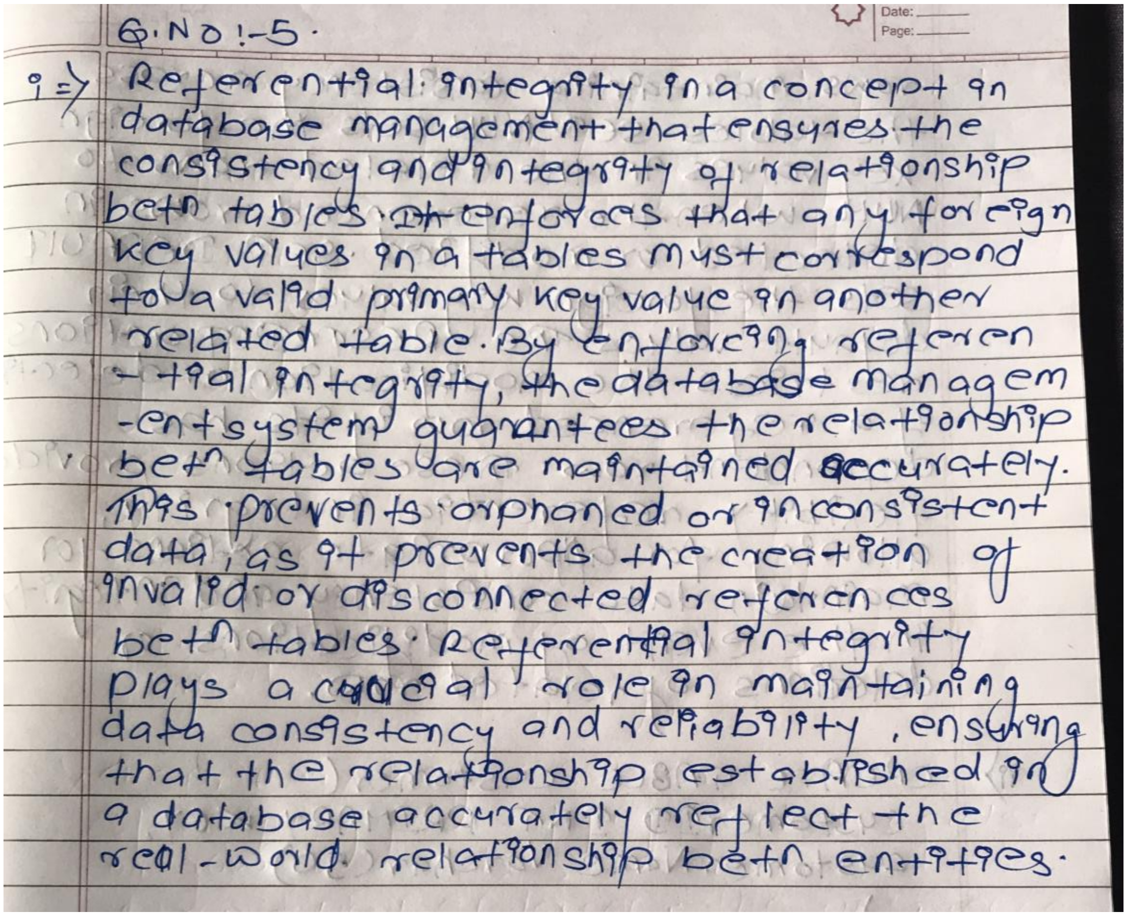
Number of rows: 25

Sort by key: None

	Employee_id	First_name	Last_name	Salary	Joining_date	Departement
□ Edit Copy Delete	2	Jerry	Kansxo	6000000	2019-01-15	IT
□ Edit Copy Delete	3	Philip	Jose	8900000	2019-02-05	Banking
□ Edit Copy Delete	4	John	Abraham	2000000	2019-02-25	Insurance
□ Edit Copy Delete	5	Michael	Mathew	2200000	2019-02-28	Finance
□ Edit Copy Delete	7	Yohan	Soso	1230000	2019-06-20	Banking

5. Establish Referential Integrity using SQL:

i. Explain the concept of referential integrity and its importance in maintaining data consistency.

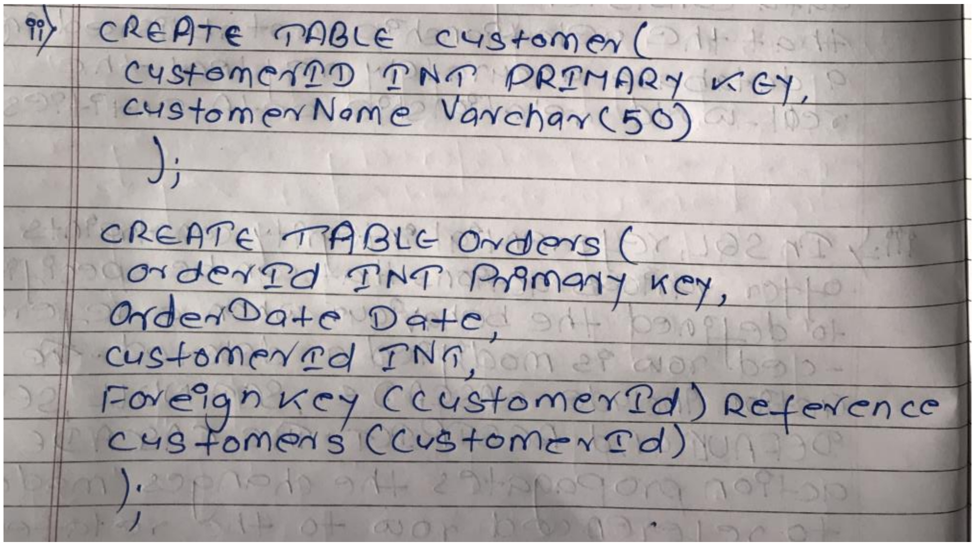


Date:
Page:

Q.No:-5.

i => Referential integrity in a concept in database management that ensures the consistency and integrity of relationship betn tables. ~~It~~ enforces that any foreign key values in a tables must correspond to a valid primary key value in another related table. By enforcing referential integrity, the database management system guarantees the relationship betn tables are maintained accurately. This prevents orphaned or inconsistent data, as it prevents the creation of invalid or disconnected references betn tables. Referential integrity plays a crucial role in maintaining data consistency and reliability, ensuring that the relationship established in a database accurately reflect the real-world relationship betn entities.

ii. Provide an example of an SQL statement to define a foreign key constraint between two tables.

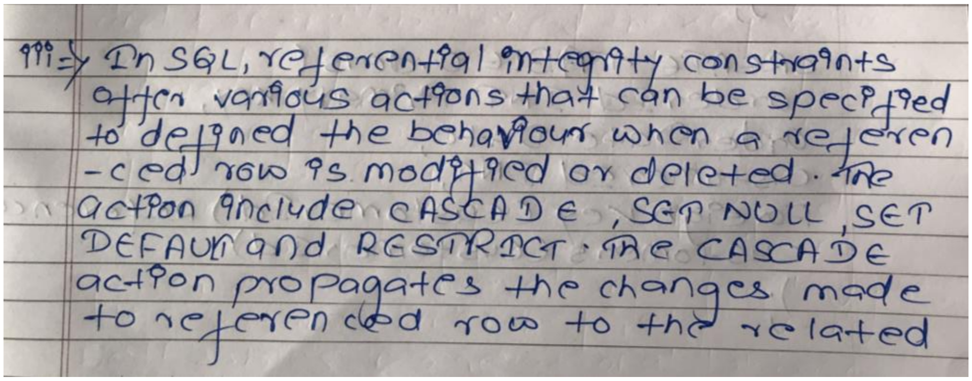


ii)

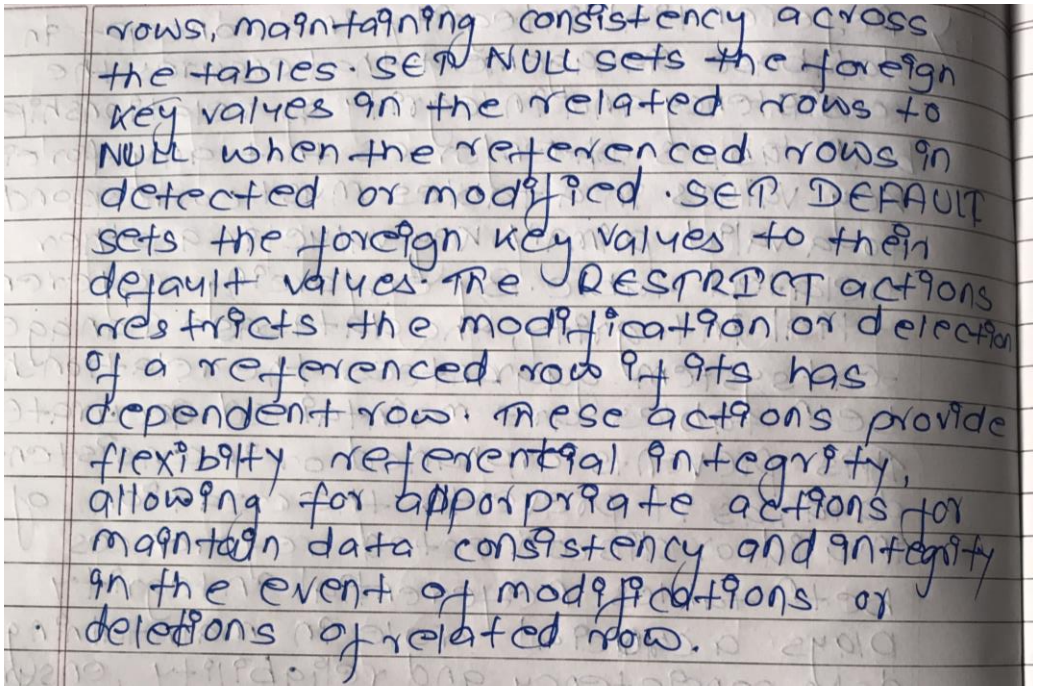
```
CREATE TABLE customer(
 customerID INT PRIMARY KEY,
 customerName Varchar(50)
   );

CREATE TABLE Orders (
 orderId INT Primary key,
OrderDate Date,
customerId INT,
Foreign key (customerId) Reference
customers (customerId)
);
```

iii. Discuss the actions that can be specified for referential integrity constraints in SQL and explain their purposes.



iii=> In SQL, referential integrity constraints offer various actions that can be specified to defined the behaviour when a referenced row is modified or deleted. The action include CASCADE, SET NULL, SET DEFAULT and RESTRICT. The CASCADE action propagates the changes made to referenced row to the related



rows, maintaining consistency across the tables. SET NULL sets the foreign key values in the related rows to NULL when the referenced rows in detected or modified. SET DEFAULT sets the foreign key values to their default values. The RESTRICT actions restricts the modification or delection of a referenced row if its has dependent row. These action's provide flexibilty referential integrity, allowing for apporpriate actions for maintain data consistency and integrity in the event of modifications or deletions of related row.

Note:

Students, for your workshop classwork, please follow the steps below:

- Import the "employee_reward.sql" file into your phpMyAdmin. This file contains the necessary database and its tables.
- Use the provided database for your classwork. Make sure to select the correct database in phpMyAdmin before executing any queries.
- Select all the tables in the database. You can do this by running the SQL query:

```
SELECT * FROM table_name;
```

- While working on your classwork or homework, make sure to properly document your work. Take screenshots of the SQL syntax you used and the output/result of your queries. This will help you keep track of your progress and serve as evidence of your work.
- Feel free to explore and apply your own ideas during the classwork or homework. SQL offers a wide range of functionalities, so don't hesitate to experiment with different queries and techniques.
- After completing the questions and answers, write your responses in your notebook or on paper.
- Once you have completed all the questions, scan each page of your work and convert them into PDF format.
- When submitting your work, include the PDF file containing your answers along with any required screenshots. This will ensure that your submission is complete and can be easily reviewed by your instructor.
- Submit your PDF file through the designated submission platform, such as the MST (Mysecond teacher nepal) platform used by your institution. Follow the provided instructions for uploading and submitting your work.

By following these guidelines, you can ensure that your work is properly documented and submitted in the required format. If you encounter any technical difficulties or have questions regarding the submission process, don't hesitate to reach out to your instructor for assistance.
Good luck with your questions and answers, and make sure to submit your work before the designated deadline.
Have fun and enjoy your workshop classwork!