SQL Session 2

Concepts to be covered:

Data Manipulation:

Data manipulation involves altering or modifying data within a database. It's a crucial aspect of managing databases as it allows users to insert, update, delete, and retrieve information to meet specific requirements. This process ensures data accuracy, consistency, and relevance, which are fundamental for making informed decisions.

Data Definition Language (DDL):

Data Definition Language (DDL) forms the backbone of database management, focusing on defining the database structure. It includes commands like **'CREATE TABLE'** to establish new tables, **'ALTER TABLE'** to modify existing table structures (like adding columns or changing data types) and establishing constraints like **Primary Keys (PK)** **or** **Foreign Keys (FK).**

The exercise you'll encounter involves hands-on experience with DDL commands. Through this exercise, you'll get to apply ALTER commands to change date field data types in tables like 'customer' or 'billing.' It's a practical way to grasp the significance of DDL in structuring and modifying databases, preparing you to handle similar tasks in real-world scenarios.

Imagine a database like a big, organized file cabinet. We'll focus on modifying how the folders inside this cabinet are set up.

Datasets Used:

- `customer`
- `billing`

Example Queries:

Data Modification Statements:

Exercise 1: Changing Data Types

Change Data Types for Date Columns:

- **Back up customer data:**

*CREATE TABLE customers_backup AS SELECT * FROM customer;*

- **Update data types:**

SET SQL_SAFE_UPDATES=0;

*-- **Subscription_Date***
UPDATE customer SET Subscription_Date = STR_TO_DATE(Subscription_Date, '%m/%d/%Y');
ALTER TABLE customer MODIFY Subscription_Date DATETIME;

*-- **Date_of_Birth***
UPDATE customer SET Date_of_Birth = STR_TO_DATE(Date_of_Birth, '%m/%d/%Y');
ALTER TABLE customer MODIFY Date_of_Birth DATETIME;

*-- **last_interaction_date***
UPDATE customer SET last_interaction_date = STR_TO_DATE(last_interaction_date, '%m/%d/%Y');
ALTER TABLE customer MODIFY last_interaction_date DATETIME;

Setting Primary Keys and Autoincremental Values:

*ALTER TABLE **customer** DROP PRIMARY KEY, ADD PRIMARY KEY (customer_id);*
ALTER TABLE customer MODIFY customer_id INT AUTO_INCREMENT;

*ALTER TABLE **billing** DROP PRIMARY KEY, ADD PRIMARY KEY (bill_id);*
ALTER TABLE billing MODIFY bill_id INT AUTO_INCREMENT;

INSERT Statements:

Inserting New Customers/Billing:

- **New customer without last interaction date:**

INSERT INTO customer (First_name, last_name, subscription_date) VALUES ("Henry", "Red", "2023-05-01");

- **Adding a new billing entry:**

INSERT INTO billing (amount_due, due_date) VALUES (100, '2023-06-10');

- **Inserting a customer with minimal details:**

INSERT INTO customer (first_name) VALUES ("Anonymous");

- **Adding billing with only the billing cycle specified:**

INSERT INTO billing (billing_cycle) VALUES ("June 2023");

UPDATE Statements:

Updating Existing Data:

- **Update customers with a subscription date before 2023-01-01:**

UPDATE customer SET last_interaction_date = '2023-05-05' WHERE subscription_date < '2023-01-01';

- **Update email for customer named "Anonymous":**

UPDATE customer SET email = 'unknown@example.com' WHERE first_name = 'Anonymous';

- **Increase late fee for overdue payments:**

UPDATE billing SET late_fee = late_fee + 5 WHERE payment_date > due_date;

- **Changing phone number for customer ID 10:**

UPDATE customer SET phone_number = '555555555' WHERE customer_id = 10;

DELETE Statements:

Deleting Records:

- **Delete customers without subscription or last interaction date:**

DELETE FROM customer WHERE subscription_date IS NULL AND last_interaction_date IS NULL;

- **Erase customers named "Anonymous":**

DELETE FROM customer WHERE first_name = 'Anonymous';

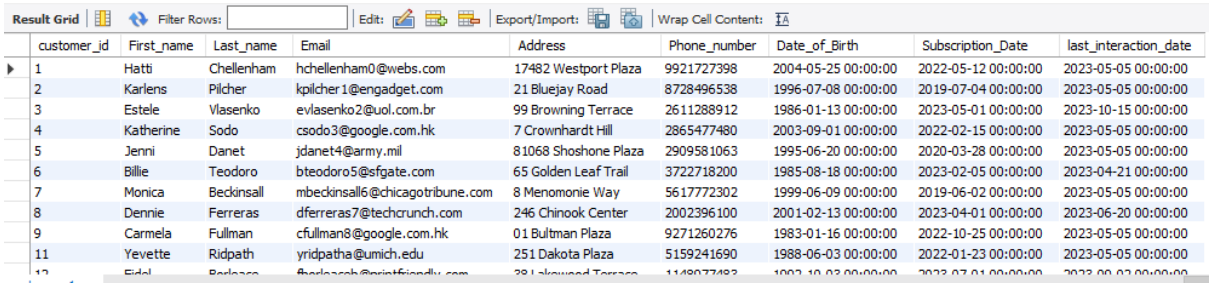
- **Deleting entries in the billing table with due date before 2022-01-01:**

DELETE FROM billing WHERE due_date < '2022-01-01';

Data Cleaning:

- **Identify customers with phone numbers not starting with "555":**

*SELECT * FROM customer WHERE phone_number NOT LIKE '555%';*



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	First_name	Last_name	Email	Address	Phone_number	Date_of_Birth	Subscription_Date	last_interaction_date
1	Hatti	Chellenham	hchellenham0@webs.com	17482 Westport Plaza	9921727398	2004-05-25 00:00:00	2022-05-12 00:00:00	2023-05-05 00:00:00
2	Karlens	Pilcher	kpilcher1@engadget.com	21 Bluejay Road	8728496538	1996-07-08 00:00:00	2019-07-04 00:00:00	2023-05-05 00:00:00
3	Estele	Vlasenko	evlasenko2@uol.com.br	99 Browning Terrace	2611288912	1986-01-13 00:00:00	2023-05-01 00:00:00	2023-10-15 00:00:00
4	Katherine	Sodo	csodo3@google.com.hk	7 Crownhardt Hill	2865477480	2003-09-01 00:00:00	2022-02-15 00:00:00	2023-05-05 00:00:00
5	Jenni	Danet	jdanet4@army.mil	81068 Shoshone Plaza	2909581063	1995-06-20 00:00:00	2020-03-28 00:00:00	2023-05-05 00:00:00
6	Billie	Teodoro	bteodoro5@sfgate.com	65 Golden Leaf Trail	3722718200	1985-08-18 00:00:00	2023-02-05 00:00:00	2023-04-21 00:00:00
7	Monica	Beckinsall	mbeckinsall6@chicagotribune.com	8 Menomonie Way	5617772302	1999-06-09 00:00:00	2019-06-02 00:00:00	2023-05-05 00:00:00
8	Dennie	Ferreras	dferreras7@techcrunch.com	246 Chinook Center	2002396100	2001-02-13 00:00:00	2023-04-01 00:00:00	2023-06-20 00:00:00
9	Carmela	Fullman	cfullman8@google.com.hk	01 Bultman Plaza	9271260276	1983-01-16 00:00:00	2022-10-25 00:00:00	2023-05-05 00:00:00
11	Yevette	Ridpath	yridpatha@umich.edu	251 Dakota Plaza	5159241690	1988-06-03 00:00:00	2022-01-23 00:00:00	2023-05-05 00:00:00

- **Replace "Road" with "Rd." in address field:**

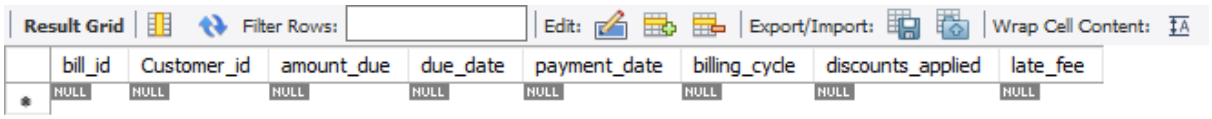
UPDATE customer SET address = REPLACE(address, 'Road', 'Rd.');

- **Convert billing cycle to uppercase:**

UPDATE billing SET billing_cycle = UPPER(billing_cycle);

- **Identify records with negative discounts applied:**

*SELECT * FROM billing WHERE discounts_applied < 0;*



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

bill_id	Customer_id	amount_due	due_date	payment_date	billing_cycle	discounts_applied	late_fee
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Remove leading/trailing whitespaces from the name field:**

UPDATE customer SET first_name = TRIM(first_name);

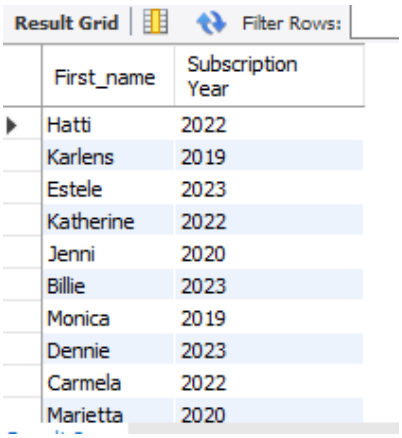
Data Transformation:

- **Adding a month to all subscription dates:**

UPDATE customer SET subscription_date = DATE_ADD(subscription_date, INTERVAL 1 MONTH);

- **Extracting the year from subscription dates:**

SELECT first_name, YEAR(subscription_date) AS 'Subscription Year' FROM customer;

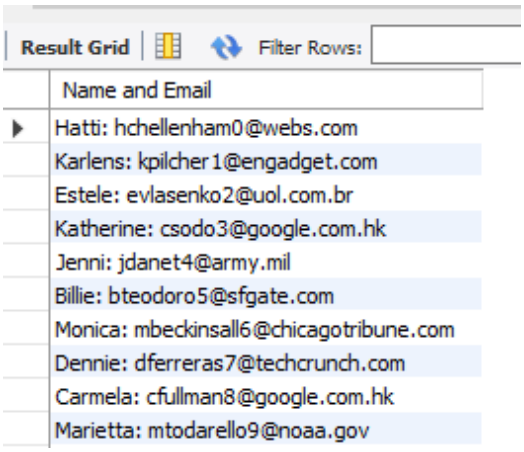


Result Grid | Filter Rows:

First_name	Subscription Year
Hatti	2022
Karlens	2019
Estele	2023
Katherine	2022
Jenni	2020
Billie	2023
Monica	2019
Dennie	2023
Carmela	2022
Marietta	2020

- **Concatenating name and email fields:**

SELECT CONCAT(first_name, ': ', email) AS 'Name and Email' FROM customer;



Result Grid | Filter Rows:

Name and Email
Hatti: hchellenham0@webs.com
Karlens: kpilcher1@engadget.com
Estele: evlasenko2@uol.com.br
Katherine: csodo3@google.com.hk
Jenni: jdanet4@army.mil
Billie: bteodoro5@sfgate.com
Monica: mbeckinsall6@chicagotribune.com
Dennie: dferreras7@techcrunch.com
Carmela: cfullman8@google.com.hk
Marietta: mtodarello9@noaa.gov

These queries perform various operations like altering data types, inserting new data, updating existing data, deleting records, cleaning data, and transforming information within the database.

Brief Explanation of concepts:

Data Type Conversion: The process of changing the format or type of data in a database. In this case, it involves altering date columns from a string format ('m/d/Y') to a date-time format.

STR_TO_DATE: Converts a string into a date using a specified format.
ALTER TABLE MODIFY: Alters the structure of a table, changing the data type for columns.

ALTER TABLE ADD PRIMARY KEY: Establishes a primary key in a table, which uniquely identifies each record.

ALTER TABLE MODIFY: Modifies the data type of a column to enable auto-incrementing values, which automatically generates unique values for new records.

INSERT Statements:

Adding New Data:
INSERT INTO: Adds new records into a table with specified values for specific columns.

UPDATE Statements:
UPDATE SET: Modifies existing records by changing specific column values that meet certain conditions**.**

DELETE Statements:
DELETE FROM: Eliminates records from a table based on specified conditions.
Data Cleaning:

Cleaning Data:
SELECT: Retrieves records from a table based on specified conditions to identify data that needs cleaning.

UPDATE SET: Modifies records to clean or standardize data. For example, replacing text strings, converting text to uppercase, or removing leading/trailing spaces.

Data Transformation:
UPDATE SET: Alters records to transform the data. For instance, adding time intervals to dates or extracting specific components (like year) from dates.

SELECT: Retrieves data to display transformed or calculated values without changing the actual stored data.

Why are these concepts important?

Data Consistency and Accuracy: Ensures that data in the database is accurate and consistent, preventing errors or discrepancies.

Database Maintenance: Allows for routine maintenance and modification of database structures to accommodate changing business needs or improve efficiency.

Data Integrity: Helps in establishing data integrity by setting primary keys, ensuring unique values, and defining relationships between tables.

Data Cleanup: Enables the correction of errors, standardization of formats, and removal of redundant or unnecessary data, maintaining data quality.

Data Transformation: Facilitates the extraction of valuable insights by transforming raw data into usable formats or aggregating information for analysis.

These concepts collectively form the backbone of managing databases, ensuring data remains organized, accurate, and accessible for effective decision-making and business operations.

Following resources offer a range of materials, from beginner-level introductions to advanced database management topics, helping you enhance your skills in SQL, database design, and data manipulation.

1.SQLZoo: Provides interactive SQL tutorials and challenges at various difficulty levels.

https://sqlzoo.net/wiki/SQL_Tutorial

2. MySQL Documentation: Official documentation for MySQL with in-depth explanations and examples.

https://dev.mysql.com/doc/refman/8.0/en/sql-statements.html

3.PostgreSQL Documentation: Official documentation for PostgreSQL database system.

https://www.postgresql.org/docs/current/ddl.html

Happy Learning! ☺