

Schemas and Tables

What is a Schema?

- Schema === Database
- A Schema is a collection of tables with rows and columns.
- A Schema will contain
 - Views
 - Tables
 - Privileges
- To create a schema, use the command 'CREATE' with its name
 - Eg. CREATE mySchema;
- To activate a schema, use the command 'USE'.
 - Eg. USE mySchema;

SQL Tables

- The schema will consist of many tables
- Data will be shown in tables of rows and columns
- Each column will have its specific data type and specific number of constraints
- We will be creating a sample table to put our data in.
- We will also be exploring different data types and constraints for our columns

Creating a Table

- There are 2 methods to create a table
 - Using the workbench UI
 - Using a query
 - Use the command 'CREATE TABLE' with the name of the table
 - Eg. CREATE TABLE users (*column names and description here*);
- Each column will also have constraints
 - Constraints are a set of rules that restrict the values inside of the table cell.
- Each column will also have specific data types which include
 - INT
 - VARCHAR
 - DATETIME
 - Etc

Modul 3

CRUD with MySQL

CRUD

- CREATE
- READ
- UPDATE
- DELETE

Creating data

- We will be creating data to be inserted to our tables inside our schema (database).
- The command 'INSERT' will be used here

Reading data (Basic)

- Fetching data from a table
 - All columns and specific columns
- We will be using the command 'SELECT'.

Reading data (filtering)

- We will be fetching specific data by writing conditions
 - Eg. username beginning with the letter 'a'
- We will be using the command 'WHERE' with 'SELECT'
- We will also be using conditional operators (AND, OR, NOT)

Reading Data (Grouping)

- Data that have the same values will be grouped into summary rows
- We will be using the command 'GROUP BY'
- This command is often used with aggregate functions such as
 - COUNT
 - MAX
 - MIN
 - SUM
 - AVG

Reading Data (sorting & limiting)

- We will be sorting the data in ascending or descending order by using the command 'ORDER BY'.
- We will be limiting the data we fetch by using the command 'LIMIT'.
- We will also be using the command 'OFFSET' to to offset the first row to return
 - Choosing at which point we will fetch our data
 - Instead of fetching at id = 1, we can fetch starting at id = 20.

Update

- We will be updating data that is already previously created
- We will be using the command 'UPDATE'

Delete

- We will be deleting existing data inside a table using the command 'DELETE'

Modul 3

Relational Database

Primary Key

- Primary keys constraint uniquely identifies each record in a database table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values
- A table can only have ONE primary key
- Examples include:
 - ID_Number / No_KTP
 - Product_Key
 - Finger_Print_ID

NDB_No	Shrt_Desc
01001	BUTTER,WITH SALT
01002	BUTTER,WHIPPED,W/ SALT
01003	BUTTER OIL,ANHYDROUS
01004	CHEESE,BLUE
01005	CHEESE,BRICK
01006	CHEESE,BRIE
01007	CHEESE,CAMEMBERT
01008	CHEESE,CARAWAY
01009	CHEESE,CHEDDAR

Composite Key

- Composite key is a combination of two or more columns in a table that can be used to uniquely identify each row in the table when the columns are combined uniqueness are guaranteed.
- Examples include
 - Name & DOB
 - Product_ID & Product_added_date

Composite Primary Key



Name	Address	City	Phone	Note
Smidth	Brooklyn	New York	18333908508	Good
Jones	Dallas	Texas	18332908508	Excellent
Lara	Orlando	Florida	18772807446	Good
Shelly	San Francisco	California	18882062011	Excellent

Foreign Key

- A foreign key is a column or field that establish the relationship between a table and another.
- Simply put, a foreign key is a primary key that is found in another table.

id	username
2	Richard
4	Andi
5	Susilo
7	Bambang
8	Joko
9	Susi
10	Bambi
11	aladin

Users

id	tanggalTransaksi	totalHarga	totalBayaran	userId
1	2019-03-22 00:00:00	100000	110000	1
2	2019-03-22 00:00:00	395000	400000	2
3	2019-03-22 14:18:21	395000	410000	3
4	2019-03-25 09:52:30	155000	160000	4
5	2019-03-25 13:11:18	160000	170000	5
6	2019-03-25 13:11:18	100000	110000	1
7	2019-03-22 00:00:00	395000	410000	2

Transaction

id	productId	harga	qty	transactionId
1	2	30000	1	1
2	1	35000	2	1
3	2	30000	10	2
4	1	35000	2	2
5	3	25000	1	2
6	1	35000	2	3
7	2	30000	10	3

Transaction_detail

Database Relationships

- When a foreign key is detected in a table, it usually mean that the table is related to another table.
- Types of these relationships include
 - One to One
 - One to Many
 - Many to Many
- Keywords : HAVING and BELONGING TO

One to One



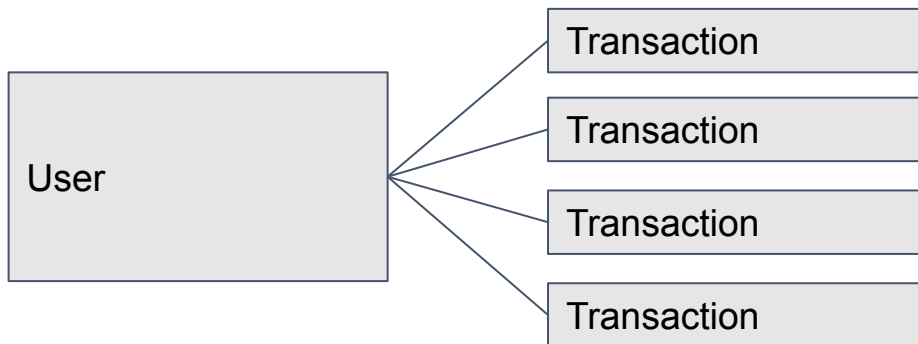
- Figure above show that a user can only have one phone number
- Said phone number can only belong to one user
- One data(field) in table B belongs to a specific field in Table A
- One user has one phone number
- The phone number belongs to one user

One to One

id	username	password
2	Richard	1
3	lianeddy	asdasd
4	Andi	1
5	Susilo	1
7	Bambang	1
8	Joko	1
9	Susi	123
10	Bambi	zxc
11	aladin	123456

id	number	user_id
1	6281234567890	2
2	6281234567891	3
3	6281234567892	4

One to Many



- One to many here refers to the user having multiple transaction data
- But one transaction data only belongs to one user
 - As two users cannot have the same transaction data

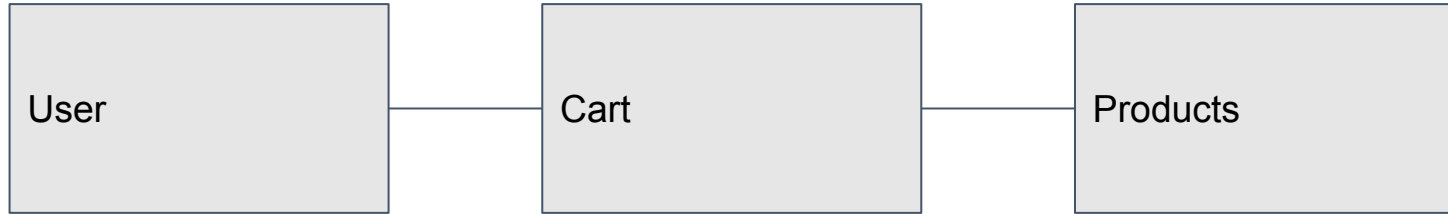
One to Many

id	username
2	Richard
4	Andi
5	Susilo
7	Bambang
8	Joko
9	Susi
10	Bambi
11	aladin

id	tanggalTransaksi	totalHarga	totalBayaran	userId
1	2019-03-22 00:00:00	100000	110000	1
2	2019-03-22 00:00:00	395000	400000	2
3	2019-03-22 14:18:21	395000	410000	3
4	2019-03-25 09:52:30	155000	160000	4
5	2019-03-25 13:11:18	160000	170000	5
6	2019-03-25 13:11:18	100000	110000	1
7	2019-03-22 00:00:00	395000	410000	2

- One user has multiple transaction data
- One transaction data BELONGS TO one user
- User (One)
- Transaction (Many)

Many to Many



- A many to many relationship occurs when multiple records in a table are associated with multiple records from another table.
- There is usually a third table to establish the relationship between the many tables

Many to Many

id	username
2	Richard
4	Andi
5	Susilo
7	Bambang
8	Joko
9	Susi
10	Bambi
11	aladin

id	userId	productId	qty
1	1	2	3
2	1	1	5
3	5	1	4
4	3	5	2
5	2	1	2
6	3	5	1
7	3	5	5

id	nama	harga
1	Paket Bento A	35000
2	Paket Bento B	30000
3	Paket Bento C	25000
4	Paket Bento D	40000
5	Paket Premium A	50000
6	Paket Premium B	55000
9	Paket Premium C	60000

- The figure above shows that one user can cart multiple products
- One product can be carted by multiple users

Database Relationships

- We will be joining tables based on the foreign keys inside of it
- We will using the command 'JOIN' & 'ON'
- We will be exploring different types of JOIN
 - INNER JOIN
 - LEFT JOIN
 - RIGHT JOIN
 - FULL OUTER JOIN

Subqueries

Definition

- A subquery is a SQL query nested inside a larger query.
- The smaller query is usually embedded within the WHERE clause
- The subquery is used to return data what will be used by the main query as a condition
- Subqueries must be closed within parentheses ().

Subqueries

- We are going to write a subquery as a where clause to our main query.
- The commands we are going to use are the 'IN' and 'NOT IN' after our 'WHERE' clause.

Example

- In the products table, we have an ID for every product
- In the transactionDetail table, we also have an ID for the product
- Using subqueries we can check for the product IDs that are not inside of the transactionDetail table, meaning products that are not yet purchased by a customer.

Modul 3

Views

What is a View?

- Whenever we generate a table with a relatively long query, eventually we are run the same query again.
- Rather than writing the same query all over again or saving it in a .txt file, we can save the query inside of our database.
- A View or a Database View is a named query stored inside of the database.

Views

- We will be creating a View using the data inside of our database.
- We will be using the command 'CREATE VIEW'

Modul 3

Event Scheduler

MySQL Events

- Events are tasks that run according to a specified schedule.
- Sometimes referred to as scheduled events
- MySQL events are named objects which contains one or more SQL statements
- The events are stored in the database to be executed at one or more intervals.
- The event scheduler manages the schedule and the execution of events

Event Scheduler

- We will be creating events using the commands 'CREATE EVENT'.
- We will be creating one-time and recurring events