HTTP5112 DATABASE

Module 8 : Database Design III

AGENDA

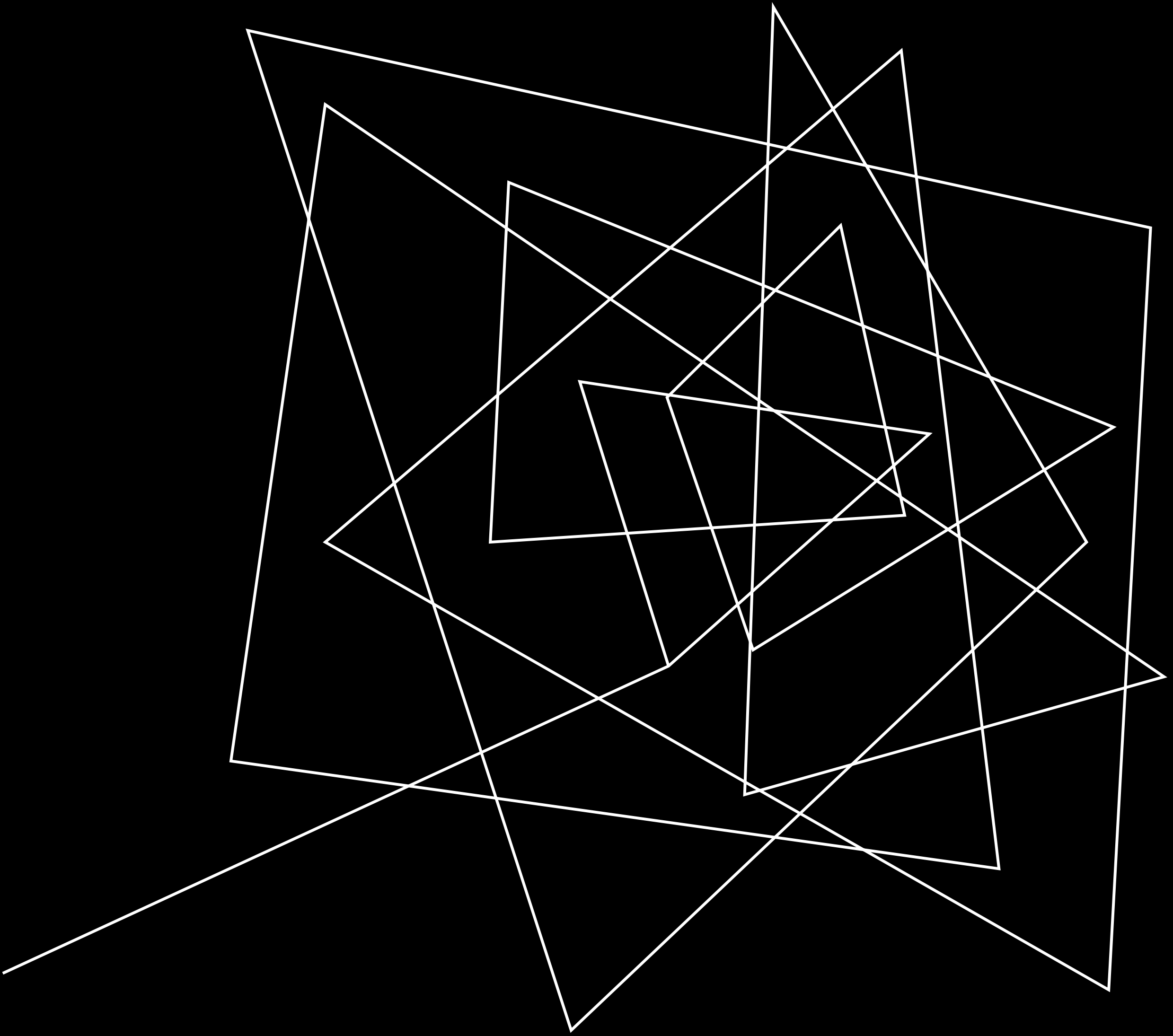
Relationship Types

Constraints

Normalization

OVERVIEW

- Data should be duplicated as little as possible
- One-to-one relationships means there are too many tables
- Many-to-many relationship means a bridging table is required
- Data integrity is maintained with constraints



RELATIONSHIP TYPES

One-to-one

One-to-Many

Many-to-Many

ONE-TO-ONE RELATIONSHIP

- User and user profile
- Student and student transcript
- Person and passport
- Library book and checkout record

EXAMPLE: ONE-TO-ONE

Student Table			Desks Table			
StudentId	Student	Desk	DeskId	Row	Column	Student
123456	Ryan	1	1	5	7	123456
123457	Amandeep	2	2	6	5	123457
123458	Ella	3	3	6	7	123458

ONE-TO-ONE: AS 1 TABLE

Seating Table			
StudentId	Student	Row	Column
123456	Ryan	5	7
123457	Amandeep	6	5
123458	Ella	6	7

ONE-TO-ONE RELATIONSHIP: WHEN TO USE?

Anticipate a one-to-many relationship

Maintain semantic distinctions between entities

Split your table if large number of number of columns

ONE-TO-MANY RELATIONSHIP

Author and Books:

1 Author can have many books, each book usually has 1 author

Department and Employees:

1 Department can have many employees, 1 employee belongs to 1 department

Customer and Orders:

1 customer can have multiple orders, each order is associated with 1 customer

Instructor and Courses:

1 instructor can teach multiple courses but each course is usually taught by 1 instructor

ONE-TO-MANY RELATIONSHIP

AUTHORS

User_id	Username	email
101	JCorden	Jcorden@mail.com
102	HL	HLee@mail.com
103	SH	ShinH@mail.com
104	MMM	MMM@mail.com

BOOKS

book_id	User_id	title
112	101	The tonight show - biography
122	101	SQL Life
123	102	Ahsoka Diaries
133	103	TVA – the history

Author and Books:

1 Author can have many books, each book usually has 1 author

MANY-TO-MANY RELATIONSHIP

Actors and Movies:

Actors can perform in multiple movies, and each movie can have multiple actors.

Employees and Projects:

Employees in an organization can work on multiple projects, and each project may involve several employees.

Customers and Products:

Customers can purchase multiple products, and products can be bought by multiple customers

Students and Courses

A student can enroll in multiple courses and each course can have multiple students

MANY-TO-MANY RELATIONSHIP: STUDENTS AND COURSE

Students

student_id	first_name	email
101	Jason	jason@mail.com
102	Mike	mike9@mail.com
103	Reena	reena@mail.com
104	Tyrone	tyrone0@mail.com

Enrollments:

Student x Courses

Enrol_id	Course_id	Student_id
1	5112	101
2	5112	102
3	5111	101
4	5211	104

Courses

Course_id	course_name
5112	Web_development Lab
5113	Special topics in web development
5111	Web_development I
5211	Web development II

CONSTRAINTS

- Rules and conditions applied to a columns to maintain data integrity, accuracy, and consistency
- Data must meet specific criteria
- Safeguards against data anomalies, errors, and inconsistencies

Primary Key

Enforces data integrity by ensuring each row in a table is unique

Foreign Key

Links two tables by specifying the values in one column (foreign key) must match values primary key in another table

CONSTRAINTS: **FOREIGN KEY – NEW TABLE**

- Create a table by assigning a foreign key that will link to the primary key (unique column) of the other table

Syntax:

```
CREATE TABLE table_1(
    column_name INT PRIMARY KEY AUTO_INCREMENT,
    foreign_key_column INT,
    FOREIGN KEY (foreign_key_column)
    REFERENCES table_2 (primary_key_column)
);
```

CONSTRAINTS: **FOREIGN KEY – EXISTING TABLE**

- Apply a foreign key constraint to an existing table

Syntax:

```
ALTER TABLE child_table
     ADD CONSTRAINT fk_foreign_key_name
     FOREIGN KEY (foreign_key_column)
     REFERENCES parent_table (primary_key_column);
```

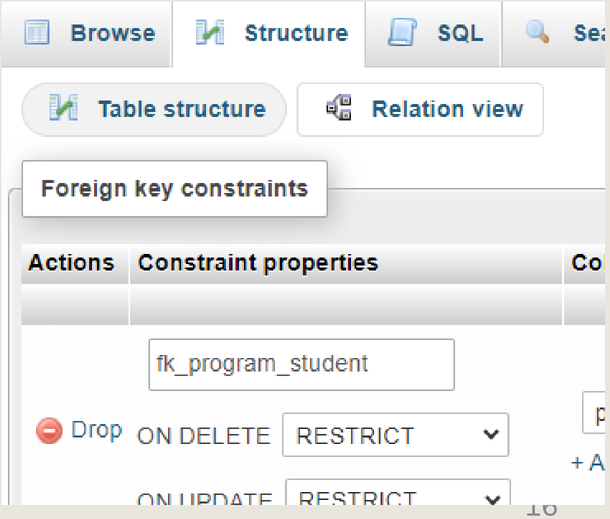
CONSTRAINTS: **FOREIGN KEY: DROP KEY**

- Drop the constraint, not column with the foreign key named by the database

Syntax:

```
ALTER TABLE table_name
    DROP FOREIGN KEY foreign_key_name;
```

- To find the specific name, click table > click “structure” tab > Relation view button >



fk_program_student

ON DELETE RESTRICT

ON UPDATE RESTRICT

NORMALIZATION

First Normal Form (1NF)

Each attribute (column) has a unique name

Domain of attributes must not change (same data type)

Each row is **uniquely identifiable** (no duplicate rows)

Each cell must have only a single value (atomicity)

Second Normal Form (2NF)

1NF criteria

Each row, outside the key, is dependent **on the entire primary key**

Third Normal Form (3NF)

2NF criteria

All non-key attributes are functionally dependent on the primary key. No transitive dependencies

UN-NORMALIZED TABLE: CUSTOMER ORDERS

Cust_Name	ProductName	Supplier	Sup_Location	Price	Brand	Order_id	email
John	Laptop	XYZ Electronics	New York	800	Dell	1	john@mail.com
Jane	Smartphone	ABC Gadgets	Los Angeles	Sa: 600 Ap: 800	Apple, Samsung	2	jane@gmail.com
John	Camera	XYZ Electronics	New York	300	Canon	3	john@gmail.com
Alice	Chair	XYZ Furniture	Boston	100	Herman Miller	4	alice@gmail.com

NORMALIZATION

First Normal Form (1NF)

Each attribute (column) has a unique name

Domain of attributes must not change (same data type)

Each row is **uniquely identifiable** (no duplicate rows)

Each cell must have only a single value (atomicity)

CHECK 1NF

Cust_Name	ProductName	Supplier	Sup_Location	Price	Brand	Order_id	email
John	Laptop	XYZ Electronics	New York	800	Dell	1	john@mail.com
Jane	Smartphone	ABC Gadgets	Los Angeles	Sa:600	Apple, Samsung	2	jane@gmail.com
John	Camera	XYZ Electronics	New York	300	Canon	3	john@gmail.com
Alice	Chair	XYZ Furniture	Boston	100	Herman Miller	4	alice@gmail.com

First Normal Form (1NF)

☑ Each column has a unique name

✖ Domain of attributes must not change (same data type)

✖ Each row is **uniquely identifiable** (no duplicate rows).

✖ Each cell must have only a single value

CHECK 1NF

Cust_Name	ProductName	Supplier	Sup_Location	Price	Brand	Order_id	email
John	Laptop	XYZ Electronics	New York	800	Dell	1	john@mail.com
Jane	Smartphone	ABC Gadgets	Los Angeles	600	Samsung	2	jane@gmail.com
John	Camera	XYZ Electronics	New York	300	Canon	3	john@gmail.com
Alice	Chair	XYZ Furniture	Boston	100	Herman Miller	4	alice@gmail.com

First Normal Form (1NF)

☑ Each column has a unique name

☑ Domain of attributes must not change (same data type)

⊗ Each row is **uniquely identifiable** (no duplicate rows).

☑ Each cell must have only a single value

1NF : SEPARATE TABLES FOR PRODUCTS, SUPPLIERS & CUSTOMER_ORDERS

Table: Products (1NF)

ProductName	Brand	Price
Laptop	Dell	800
Smartphone1	Apple	600
Smartphone2	Samsung	800
Camera	Canon	300
Chair	Herman Miller	100

Table: Customer_orders (1NF)

Order_id	Cust_Name	email
1	John	john@mail.com
2	Jane	jane@gmail.com
3	John	john@gmail.com
4	Alice	alice@gmail.com

Table: Suppliers (1NF)

Supplier	Location
XYZ Electronics	New York
ABC Gadgets	Los Angeles
XYZ Furniture	Boston

First Normal Form (1NF)

- ☑ Each column has a unique name
- ☑ Domain of attributes must not change
- ☑ Each row is **uniquely identifiable**
- ☑ Each cell must have only a single value

1NF : ADD PRIMARY KEYS

Table: Products (1NF)

Prod_id	ProductName	Brand	Price
1	Laptop	Dell	800
2	Smartphone1	Apple	600
3	Smartphone2	Samsung	800
4	Camera	Canon	300
5	Chair	Herman Miller	100

Primary key

Table: Customer_orders (1NF)

Order_id	Cust_Name	email
1	John	john@mail.com
2	Jane	jane@gmail.com
3	John	john@gmail.com
4	Alice	alice@gmail.com

Table: Suppliers (1NF)

Sup_id	Supplier	Location
10	XYZ Electronics	New York
12	ABC Gadgets	Los Angeles
13	XYZ Furniture	Boston

First Normal Form (1NF)

☑ Each column has a unique name

☑ Domain of attributes must not change

☑ Each row is **uniquely identifiable**

☑ Each cell must have only a single value

NORMALIZATION

Second Normal Form (2NF)

1NF criteria

Each row, outside the key, is dependent **on the entire primary key**

SECOND NORMAL FORM :2NF

Table: Customer_orders

Order_id	Cust_Name	email
1	John	john@mail.com
2	Jane	jane@gmail.com
3	John	john@gmail.com
4	Alice	alice@gmail.com

Second Normal Form (2NF)

☑ 1NF criteria

✖ Each row, outside the key, is dependent **on the entire primary key**

2NF : SEPARATE TABLES FOR PRODUCTS, SUPPLIERS & CUSTOMER_ORDERS

Table: Customer_orders

Order_id	Cust_id
1	1
2	2
4	1
5	3

Table: Customers

cust_id	Cust_Name	email
1	John	john@mail.com
2	Jane	jane@gmail.com
3	Alice	alice@gmail.com

Second Normal Form (2NF)

☑ 1NF criteria

☑ Each row, outside the key, is dependent **on the entire primary key**

NORMALIZATION

Third Normal Form (3NF)

2NF criteria

All non-key attributes are functionally dependent on the primary key. No transitive dependencies

THIRD NORMAL FORM: 3NF

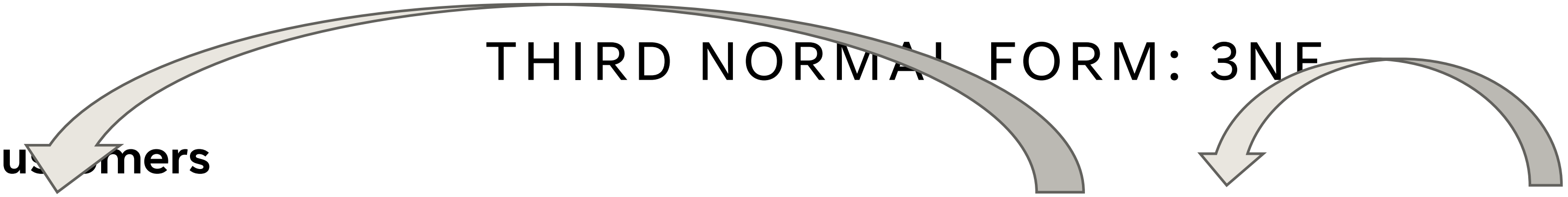


Table: Customers

cust_id	Cust_Name	email	Address	city
1	John	john@mail.com	1 Front Street	Toronto
2	Jane	jane@gmail.com	2 Bloor Street	Toronto
3	Alice	alice@gmail.com	3 Lakeshore Blvd	Toronto
4	Min	minlee1@gmail.com	8 Beach Ave	Vancouver
5	Jay	jay33@gmail.com	1 Georgia Street	Vancouver
6	Sam	sam112@gmail.com	10 Yonge Street	Toronto

Third Normal Form (3NF)

☑ 2NF criteria

✗ All non-key attributes are functionally dependent on the primary key. No transitive dependencies

THIRD NORMAL FORM: 3NF

Table: Customers

cust_id	Cust_Name	email	Address	City_id (FK)
1	John	john@mail.com	1 Front Street	1
2	Jane	jane@gmail.com	2 Bloor Street	1
3	Alice	alice@gmail.com	3 Lakeshore Blvd	1
4	Min	minlee1@gmail.com	8 Beach Ave	2
5	Jay	jay33@gmail.com	1 Georgia Street	2
6	Sam	sam112@gmail.com	10 Yonge Street	1

Table: City

city_id (PK)	city
1	Toronto
2	Vancouver

Third Normal Form (3NF)

☑ 2NF criteria

☑ All non-key attributes are functionally dependent on the primary key. No transitive dependencies

SUMMARY:

- Normalization of a dataset creates relational database
- 3 levels of Normalization (1NF, 2NF, 3NF): data should be duplicated as little as possible
- One-to-one relationships means there are too many tables
- Many-to-many relationship means a bridging table is required
- Data integrity is maintained with constraints – foreign key

THIS WEEK:
ASSIGNMENT 8