

# PLEASE COMPLETE THE SOCRATIVE Pulse Survey

URL: [gosocrative.com](https://gosocrative.com)

ROOM NAME: JAVAGOLD

# Database Design

Module 2: 05

# Week 5 Overview

**Monday**

Database  
Design

**Tuesday**

Data Access  
Part 1

**Wednesday**

Data Access  
Part 2

**Thursday**

Data Access  
Security and  
Testing

**Friday**

Review

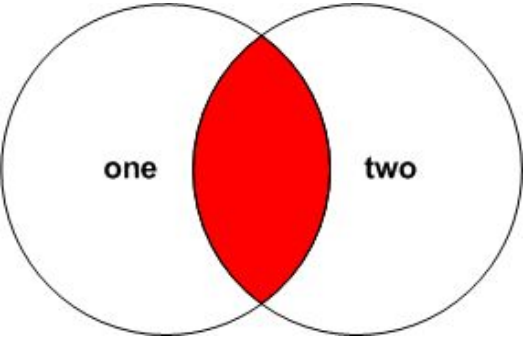
# Objectives

1. Database Definition Language` (DDL)
  - a. Working with Databases and Tables
  - b. Sequences
  - c. Schema
2. Database Control Language (DCL)
  - a. Users and Permissions
3. Normalization

# Inner Join

(Default)

```
SELECT one.number AS one_number,  
one.description as one_description, two.number  
as two_number, two.description as  
two_description  
FROM one  
JOIN two ON one.number = two.number
```

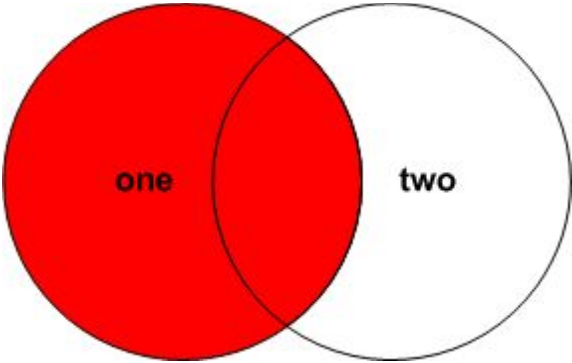


Tables joined on number			
on.number	one.description	two.number	two.description
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

# Left Join

( Left Outer Join )

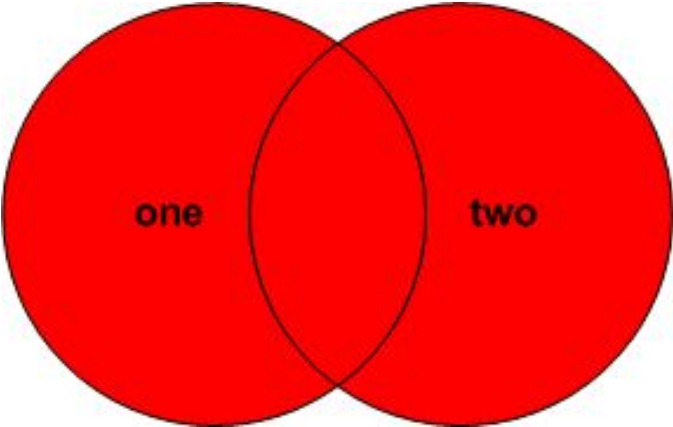
```
SELECT one.number AS one_number,  
one.description as one_description, two.number  
as two_number, two.description as  
two_description  
FROM one  
LEFT JOIN two ON one.number = two.number
```



Tables joined on number			
on.number	one.description	two.number	two.description
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

# Full Outer Join

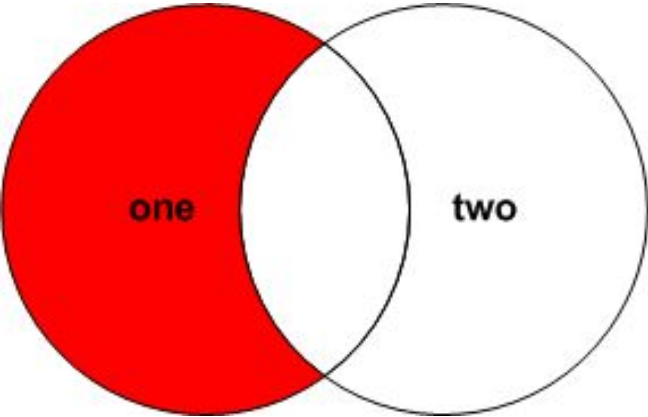
```
SELECT one.number AS one_number,  
one.description as one_description, two.number  
as two_number, two.description as  
two_description  
FROM one  
FULL OUTER JOIN two ON one.number = two.number
```



Tables joined on number			
on.number	one.description	two.number	two.description
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

# Only the Left Table Values (Unnamed)

```
SELECT one.number AS one_number,  
one.description as one_description, two.number  
as two_number, two.description as  
two_description  
FROM one  
LEFT JOIN two ON one.number = two.number  
WHERE two.number IS NULL
```



Tables joined on number			
on.number	one.description	two.number	two.description
100	ONE - 100	null	null
101	ONE - 101	null	null
102	ONE - 102	null	null
103	ONE - 103	null	null
104	ONE - 104	null	null
105	ONE - 105	null	null
990	ONE-BOTH - 990	990	TWO-BOTH - 990
991	ONE-BOTH - 991	991	TWO-BOTH - 991
992	ONE-BOTH - 992	992	TWO-BOTH - 992
993	ONE-BOTH - 993	993	TWO-BOTH - 993
994	ONE-BOTH - 994	994	TWO-BOTH - 994
995	ONE-BOTH - 995	995	TWO-BOTH - 995
null	null	200	TWO - 200
null	null	201	TWO - 201
null	null	202	TWO - 202
null	null	203	TWO - 203
null	null	204	TWO - 204
null	null	205	TWO - 205

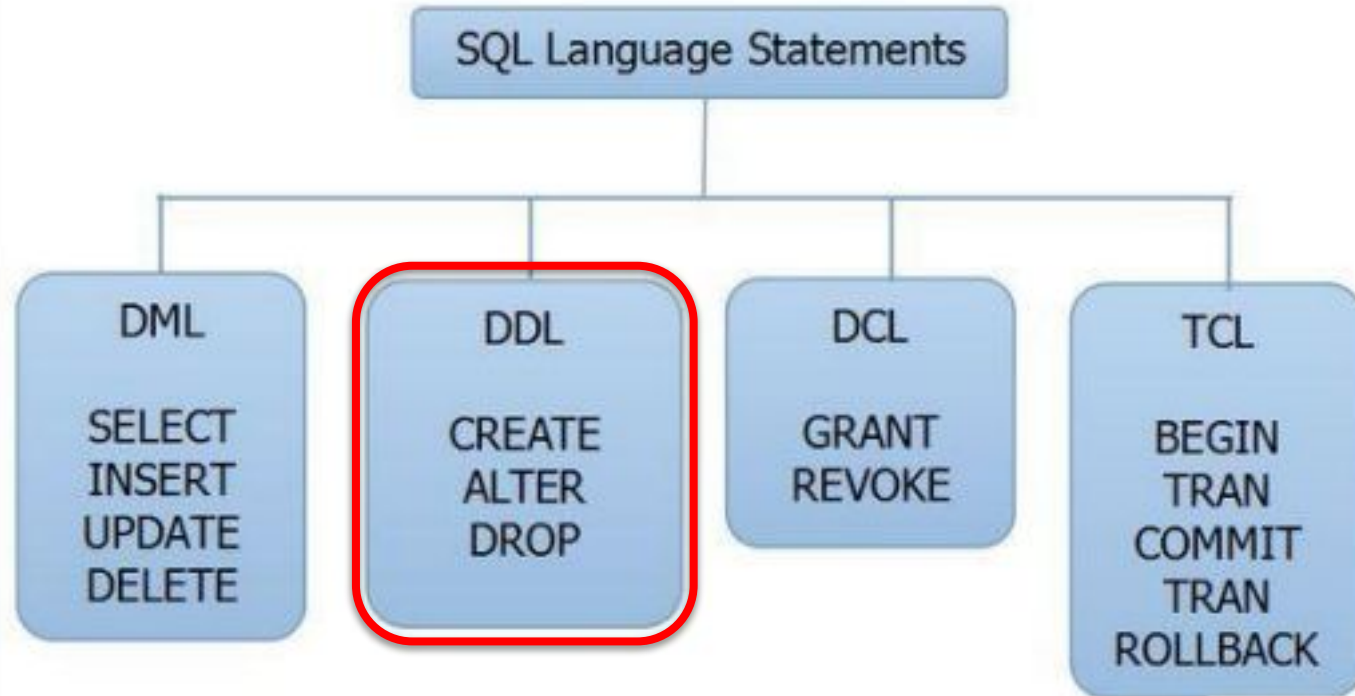


DML - Data Manipulation Language

DDL - Data Definition Language

DCL - Data Control Language

TCL - Transaction Control Language



# DQL vs DML vs DDL

The SQL statements we have seen so far fall into a number of different categories:

- Data Query Language (**DQL**): SELECT
- Data Manipulation Language (**DML**): INSERT, UPDATE, DELETE
- Data Definition Language (**DDL**): CREATE, ALTER

The focus of this lecture will be DDL statements with appropriate constraints.

# DDL (Database Definition Language)

DDL

CREATE

ALTER

DROP

A sublanguage of SQL that consists of a set of commands used to define the database, tables, indexes, keys, constraints, and other metadata that comprise a database ***schema***.

A ***schema*** is a skeletal structure of the database that focuses on the tables and constraints without the data.

# DDL Database Commands

## Creating a Database

```
CREATE DATABASE database_name;
```

## Deleting a Database

```
DROP DATABASE database_name;
```

**DROP** is the term used by DDL to indicate that something is being removed.

**Data** from the table *is DELETED*

**Structure**, like a database, table, column, or sequence, *is DROPPED*

# Creating Tables

The CREATE TABLE statement creates a new table and defines its structure. The columns and data types are required. The Primary Key, Foreign Keys, and other constraints are typically included, but can be added later.

```
CREATE TABLE table_name (  
    column_name1 data_type(size),  
    column_name2 data_type(size) NOT NULL,  
    column_name3 data_type,  
    CONSTRAINT pk_column_1 PRIMARY KEY (column_name1),  
    CONSTRAINT fk_column_2 FOREIGN KEY (column_name2) REFERENCES table_name(column_1)  
);
```

- The **column name** specifies the exact name of the column.
- The **data type** indicates what it holds.
- The **size** specifies the maximum length of the column, when required by the data type.

# Changing Tables

DROP TABLE removes a table.

```
DROP TABLE table_name
```

ALTER TABLE can change the structure of a table or add a constraint.

```
ALTER TABLE table_name ADD CONSTRAINT pk_constraint_name PRIMARY KEY (column_name(s));
```

```
ALTER TABLE table_name ADD CONSTRAINT fk_constraint_name FOREIGN KEY (column_name) REFERENCES  
table(column_name);
```

```
ALTER TABLE table_name ADD CONSTRAINT chk_constraint_name CHECK (column_name = 'value' OR column_name  
IN (values));
```

[Alter Table Documentation](#)

# Sequences

Sequences are incrementing numbers that are commonly used as Surrogate Primary Keys. Start at 0, unless given a starting value. Never stops incrementing.

Creating a Sequence manually:

```
CREATE SEQUENCE custom_seq;
```

Getting the next number manually:

```
SELECT nextval('custom_seq');
```

Sequences are not affected by a *rollback* of a transaction.

Creating a when creating a table:

```
column_name serial
```

Getting the next number automatically:

```
INSERT... (serial_col) VALUES (DEFAULT);
```

Or don't include the column in the Insert, and it will create and populate it automatically.

# DCL (Database Control Language)

Database Control Language (DCL) is used to administer the database, users, and permissions.

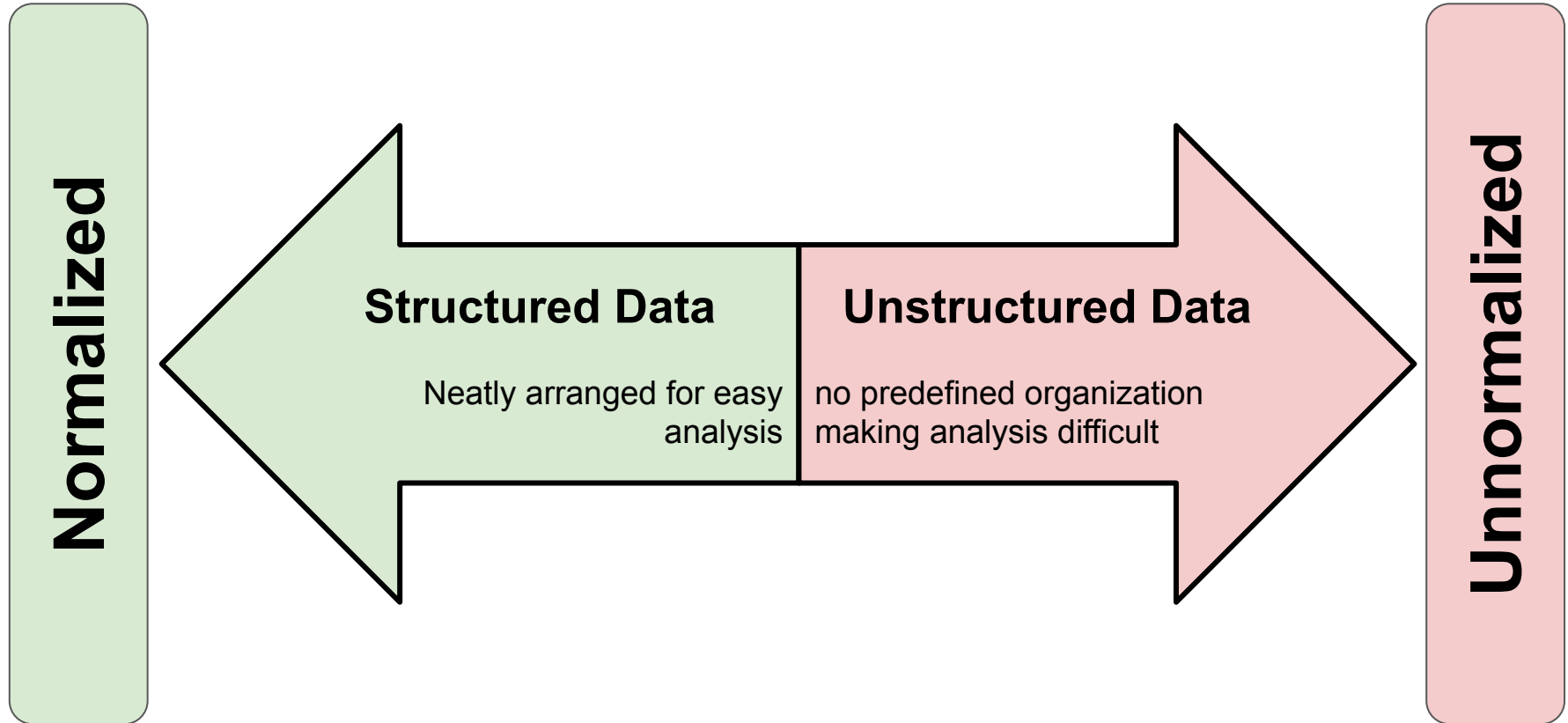
**GRANT** - gives access to a specific action for a resource to a user.

**REVOKE** - removes access to specific action for a resource from a user.





# Database Normalization



# Normalization

The process of improving a database design in steps, called forms of normalization.

Normalization has 4 goals:

1. arranging data into logical groupings such that each group describes a small part of the whole;
2. minimizing the amount of duplicate data stored in a database;
3. organizing the data such that, when you modify it, you make the change in only one place
4. building a database in which you can access and manipulate the data quickly and efficiently without compromising the integrity of the data in storage.

# Normal Forms

Before a single CREATE statement is run, the tables and their relationships need to be well thought out.

Normalization is the process of organizing a database to reduce data redundancy and improve data integrity.

We normalize data to:

1. Avoid duplicate data
2. Fix anomalies
3. Simplify search queries.

# Normal Forms: 3NF

There are several levels above 3NF of “normal form” compliance, but generally the third normal form is good enough for 99% of all situations.

An informal intuitive definition of 3NF is as follows:

There are no fields in a table that are not directly determined by the values of the primary key.

Therefore, all fields in a table should be directly related to (determined by) the primary key of that table.

**Gallery Customer History Form**

Customer Name

Jackson, Elizabeth  
123 – 4<sup>th</sup> Avenue  
Fonthill, ON  
L3J 4S4

Phone (206) 284-6783

**Purchases Made**

Artist	Title	Purchase Date	Sales Price
03 - Carol Channing	Laugh with Teeth	09/17/2000	7000.00
15 - Dennis Frings	South toward Emerald Sea	05/11/2000	1800.00
03 - Carol Channing	At the Movies	02/14/2002	5550.00
15 - Dennis Frings	South toward Emerald Sea	07/15/2003	2200.00

The Gill Art Gallery wishes to maintain data on their customers, artists and paintings. They may have several paintings by each artist in the gallery at one time. Paintings may be bought and sold several times. In other words, the gallery may sell a painting, then buy it back at a later date and sell it to another customer.