A close-up of a certificate

Description automatically generated

[Introduction to Databases for Back-End Development - Introduction to Databases - Week 1 | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/home/week/1) 3-22-2024 0942

[Introduction to databases | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/RcZYr/introduction-to-databases) 3-22-2024 1200

* Short summary: This course introduces databases, their importance in various industries, and the fundamental actions that databases perform. You'll learn key database concepts and build your own database during the course.
* Detailed Breakdown
* Acronyms with definitions
  + IOT: Internet of Things – the network of physical devices, vehicles, and other items connected to the internet and collecting data.
* Technical words with definitions
  + Database: An organized collection of data, generally stored and accessed electronically.
  + Data: Facts and statistics collected together for reference or analysis.
  + Big Data: Extremely large and complex datasets that require specialized technologies to analyze and process.
  + Filter: To process data and show only records that meet specific criteria.
  + Search: To look for specific records within a database that match certain parameters.
  + Update: To change existing data within a database.
  + Delete: To remove data from a database.
  + Back-end developer: A programmer who works on the server-side of applications, responsible for building databases and the logic that powers websites and software.
  + Rarely used words (for a 12-year-old) with definitions
  + Diverse: Showing a great deal of variety.
  + Accommodate: To provide with something needed.
  + Revolution: A major, sudden change in a situation or way of thinking.
  + Segmented: Divided into separate sections or parts.
  + Normalize: To bring something into conformity with a standard or norm.
  + Software packages mentioned with definitions
  + WAMP: (Likely referring to a software stack) A combination of software including Windows (operating system), Apache (web server), MySQL (database system), and PHP (programming language).
* Ideas mentioned in the transcript with definitions
  + Database Structure: The way a database is organized to store and relate data.
  + Create, Read, Update, Delete (CRUD): The fundamental operations performed on data within a database.
  + SQL Operators: Commands used in SQL (Structured Query Language) to manipulate data, including sorting and filtering.
  + Database Normalization: The process of structuring a database to reduce redundancy and improve data integrity.
* Keywords keywords = ["database", "data", "big data", "developer", "CRUD", "SQL", "normalization"]
* Written by: Gemini

[A day in the Life of a Database Engineer | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/Bv0R3/a-day-in-the-life-of-a-database-engineer) 3-22-2024 1215

* Day in the Life of a Database Engineer

[What is a database? | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/bS041/what-is-a-database) 3-2024 1220

* Super basic

[How is data related? | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/lWFd7/how-is-data-related) 3-22-2024 1245

* Short summary: This transcript explains how databases form relationships between different pieces of data using primary and foreign keys. This is essential for organizing information and retrieving it in a meaningful way.
* Technical words
* Database: An organized collection of data, generally stored and accessed electronically.
* Table: A structure within a database that holds data, organized into rows and columns.
* Relational Database: A type of database where data is organized into tables and relationships can be established between them.
* Terms in transcript
* Field: A column in a database table, representing a specific type of data (e.g., customer name, order ID).
* Record: A row in a database table, containing individual data points for all the fields.
* Entity: A real-world object or concept that is represented by data in the database (e.g., a customer, an order).
* Primary Key: A field in a database table whose values uniquely identify each record in that table.
* Foreign Key: A field in a database table that links to the primary key of another table, establishing a relationship.
* Table: The fundamental structure within a database. A table organizes data into rows (representing individual records) and columns (representing fields or attributes).
* Instance: A specific occurrence or example of something. In databases, an instance refers to a single record or row within a table. For example, a customer with the name "John Smith" is an instance of the "Customer" entity.
* Ideas mentioned in the transcript
  + Database Relationships: The way different tables in a database are connected to each other through primary and foreign keys.
* Keywords keywords = ["database", "table", "relational database", "field", "record", "entity", "primary key", "foreign key"]
* Written by: Gemini

[Relational data example charts | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/IXwQs/relational-data-example-charts) 3-22-2024 1330

* Short summary: This transcript introduces different types of charts used to visually represent data. It highlights the importance of choosing the appropriate chart to effectively communicate information to an audience.
* COVID-19: Coronavirus Disease 2019 (a highly infectious respiratory illness)
* Ideas mentioned in the transcript with definitions
* Data Visualization: The graphical representation of data to aid in understanding patterns or trends.
* Keywords keywords = ["data", "chart", "bar chart", "bubble chart", "line chart", "pie chart", "visualization"]
* Written by: Gemini

[Alternative types of databases | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/uhNBa/alternative-types-of-databases) 3-22-2024 1333

* Summary: This video explores how databases have evolved to handle modern challenges. It discusses the limitations of traditional relational databases, the rise of NoSQL databases for unstructured data, the impact of big data, cloud databases, and business intelligence technologies
* Acronyms
  + IoT: Internet of Things
* Terms
* Databases: Organized collections of data
* Structured Data: Data organized into a predefined format (tables, rows, columns)
* Unstructured Data: Data without a predefined format (text, images, etc.)
* Technical Words
* Big Data: Extremely large and complex datasets
* Cloud Databases: Databases stored and accessed through cloud computing networks
* Business Intelligence (BI): Technologies and strategies to analyze data for decision-making
* Database Types: Relational vs. NoSQL and their uses
* Big Data Characteristics: Large volume, complexity, and growth over time
* Cloud Database Benefits: Scalability, cost-effectiveness
* NoSQL database techniques: For handling unstructured data effectively
* keywords = ['databases', 'structured data', 'unstructured data', 'relational databases', 'NoSQL databases', 'big data', 'cloud databases', 'business intelligence']

This analysis was provided by Gemini, a large language model from Google AI.

[Database Evolution | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/cXSl1/database-evolution) 3-22-2024 1417

* Summary: This reading charts the history of database technology. It explains how databases evolved from early file-based systems to modern object-oriented and NoSQL databases, driven by the need to store and process different data types.
* Acronyms
  + JSON: JavaScript Object Notation
  + DBMS: Database management system (DBMS): A comprehensive software system for defining, creating, and using databases
* Terms
  + Structured data: Data organized in a predefined format (tables, rows, columns)
  + Unstructured data: Data that doesn't follow a predefined format (ex: social media posts)
  + Big Data: Extremely large, complex, and rapidly growing datasets
  + Table: A collection of data organized into rows and columns within a database
* Technical Words
  + Query language: A programming language for retrieving and manipulating data in a database
  + Primary key: A unique identifier for each record in a database table
  + Foreign key: A column in a database table referencing the primary key of another table, creating a relationship
  + Object-oriented programming: Programming based on representing data and behaviors as objects
* Rarely Used Words
  + Paradigm: A typical pattern or model
  + Ad-hoc: Created or done for a particular purpose as necessary
  + Semi-structured data: Data with some organization, but not a fully rigid structure
* Commands
  + SQL (Structured Query Language): A standard language for interacting with relational databases
* Ideas
  + Database models: Different ways of structuring and organizing data in a database (flat file, hierarchical, network, relational, object-oriented, NoSQL)
  + Relational database design: The process of creating a structure for a relational database with tables, relationships, and constraints
  + NoSQL database features: Scalability, flexibility, and the ability to handle unstructured data
* Techniques
  + Data modeling: The process of creating a visual representation of data structures and their relationships
* Important People
* E.F. Codd: Inventor of the relational database model
* Charles Bachmann: Developer of the network database model
* Important Dates
  + 1960s: Emergence of computerized databases
  + 1970s-1990s: Flat files, hierarchical, and network databases
  + 1980s: Introduction of relational databases
  + 1990s: Object-oriented and NoSQL databases
* Theories
  + Relational database theory: The mathematical foundation for organizing data in relational databases
* keywords = ['database', 'database models', 'flat file', 'hierarchical database', 'network database', 'relational database', 'SQL', 'object-oriented database', 'NoSQL', 'structured data', 'unstructured data', 'big data']

This analysis was provided by Gemini, a large language model from Google AI.

[Knowledge check: Databases and data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/1c27d/knowledge-check-databases-and-data/attempt?redirectToCover=true) 3-22-2024 1423

The foreign key can be used to identify a specific record of data in a relational database. The was a true or false question that the quiz said is false, but I don’t get it. A foreign key identifies a specific record in a database, its just in a different table.

[What is Structured Query Language? | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/xbQX5/what-is-structured-query-language) 3-22-2024 1428

Just an introduction

CRUD: **C**reate, **R**ead, **U**pdate and **D**elete

[SQL usage | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/G3a1M/sql-usage) 3-22-2024 1439

* Summary: This reading explains how SQL is used to manage databases. It covers CRUD operations (create, read, update, delete) and the different SQL sublanguages (DDL, DML, DQL, DCL) that perform specific database tasks.
* Acronyms
  + CRUD: Create, Read, Update, Delete
  + DDL: Data Definition Language
  + DML: Data Manipulation Language
  + DQL: Data Query Language
  + DCL: Data Control Language
* Terms
  + Database: An organized collection of data
  + Table: A structure within a database to store data in rows and columns
  + SQL: Structured Query Language, a standard language for interacting with databases
* Commands
  + CREATE: (DDL) Used to create database objects like tables
  + ALTER: (DDL) Used to modify the structure of existing database objects
  + DROP: (DDL) Used to remove database objects
  + INSERT: (DML) Used to add new data into a table
  + UPDATE: (DML) Used to modify existing data in a table
  + DELETE: (DML) Used to remove data from a table
  + SELECT: (DQL) Used to retrieve data from a database
  + GRANT: (DCL) Used to give users permissions to access data
  + REVOKE: (DCL) Used to remove user permissions to access data
* Ideas
  + CRUD Operations: Fundamental actions for interacting with data in a database
  + SQL Sublanguages: Specialized parts of SQL for specific tasks (defining data, manipulating data, querying data, controlling access)
* keywords = ['SQL', 'database', 'CRUD', 'DDL', 'DML', 'DQL', 'DCL', 'table', 'data', 'query']

This analysis was provided by Gemini, a large language model from Google AI.

[Advantages of SQL | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/PZSVH/advantages-of-sql) 3-22-20241426

* Summary: This reading highlights the advantages of using SQL for database interactions. It emphasizes SQL's simplicity, standardization, portability, comprehensiveness, and efficiency in handling large datasets.
* Acronyms
  + CRUD: Create, Read, Update, Delete
  + SQL: Structured Query Language
  + DDL: Data Definition Language
  + DML: Data Manipulation Language
  + DQL: Data Query Language
  + DCL: Data Control Language
* Software Packages
  + MySQL: A popular open-source relational database management system
* keywords = ['SQL', 'database', 'relational database', 'CRUD', 'DDL', 'DML', 'DQL', 'DCL', 'query', 'portability']

This analysis was provided by Gemini, a large language model from Google AI.

[SQL syntax introduction | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/BlJ3x/sql-syntax-introduction) 3-22-2024 1453

* Summary: This reading introduces basic SQL syntax and its sublanguages (DDL, DML, DQL) for database interactions. It demonstrates how to use SQL for creating a database, adding/modifying data, and querying information.

[Knowledge check: SQL syntax review | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/AYW4j/knowledge-check-sql-syntax-review) 3-22-2024 1626

* A little bit of an issue, but not worth coming back to.

[What are tables in databases? | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/7VpMb/what-are-tables-in-databases) 3-22-2024 1633

* Basic stuff

[Tables overview | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/UjZNO/tables-overview)

* Summary: This reading explains how relational databases use tables to store data. It covers columns, rows, data types, keys (primary and foreign), and constraints that ensure data integrity.
* All Acronyms in Transcript with Definitions
  + SDLC: Software Development Life Cycle (A framework outlining stages involved in building software).
  + SQL (Structured Query Language): The standard language for interacting with relational databases.
* All Terms in Transcript with Definitions
  + Database: An organized collection of data, typically stored electronically.
  + Relational Database: A database that organizes data into tables with relationships between them.
  + Table: The basic unit of data storage in a relational database, consisting of rows and columns.
  + Primary Key: A unique identifier for each row in a table.
  + Foreign Key: A column in a table that references a primary key in another table, establishing a relationship between them.
  + Integrity Constraints: Rules that enforce data consistency and correctness in a database.
* All Technical Words in Transcript with Definitions
  + Structure: The arrangement or organization of elements within a table or database.
  + Binary data: Data represented in a form computers can understand (using 0s and 1s).
  + Schema: The blueprint of a database table, defining its name, columns, and their data types.
  + Encode: Convert data into a specific format.
* All Rarely Used Words a 12-year-old Might Not Understand
  + Tuple: A technical term for a row in a database table.
* keywords = ['database', 'relational database', 'table', 'row', 'column', 'data type', 'SQL', 'primary key', 'foreign key']

Written by Gemini

[Database structure overview | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/7MqYr/database-structure-overview) 3-23-2024 0545

Summary: This reading offers a fundamental overview of database structures, focusing on tables, fields, records, keys, and their interrelationships within a database. It emphasizes the importance of data types and the role of primary and foreign keys in linking tables together.

Acronyms:

* ERD: Entity Relationship Diagram
* DBMS: Database Management System
* CLOB: Character Large Object
* BLOB: Binary Large Object
* INT: Integer
* CHAR: Character
* VARCHAR: Variable Character

Technical Words with Definition:

* Database: A structured set of data held in a computer.
* Table: A collection of related data held in a structured format within a database.
* Attribute: A quality or feature regarded as a characteristic or inherent part of something.
* Field: A single piece of data in a record, typically representing a particular attribute.
* Record: A collection of fields that typically represents a single entity.
* Primary Key: A unique identifier for a record in a table.
* Foreign Key: A field in one table that uniquely identifies a row of another table.
* Data Type: A classification identifying one of various types of data, such as integer, floating-point, or string, determining the operations that can be performed on the data.

Rarely Used Words:

* Tuple: A record in a table.
* Cardinality: In the context of databases, it refers to the uniqueness of data values contained in a column.

Software Packages:

* MySQL: A relational database management system based on SQL – Structured Query Language.
* Oracle: A multi-model database management system primarily designed for enterprise grid computing.

Commands/Modules:

* SQL: Structured Query Language, a standard programming language for managing and manipulating databases.

Ideas/Techniques with Definition:

* Entity Relationship Diagram (ERD): A diagram that shows the relationship between different entities in a database system.
* Logical Database Structure: The abstract design of a database's data model, which includes the ERD and the logical interrelationships between entities.
* Physical Database Structure: The actual implementation of the database on a storage system, detailing how data is stored and accessed.

Keywords: database structure, tables, fields, records, keys, data types, ERD, DBMS, primary key, foreign key

Written by ChatGPT

[Types of keys in a database table | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/pFwgX/types-of-keys-in-a-database-table) 3-23-2024 0624

Summary: This video explains the foundational concepts of the relational database model, focusing on entities, relations, and the various types of keys used to establish relationships between tables in a database. It illustrates these concepts using a sports competition database example, detailing how keys like primary, foreign, and composite keys function within this framework.

* Acronyms: ID: Identification
* Technical Words with Definition:
  + Relational Database Model: A type of database model that organizes data into tables (or relations) that can be linked—or related—based on data common to each.
  + Entities: Tables in a relational database that store data about objects, people, or concepts.
  + Relations: Connections between entities in a database, established through keys.
  + Key Attribute: A specific column in a table designed to uniquely identify each row or record.
  + Candidate Key: An attribute or set of attributes that uniquely identifies a row in a table.
  + Composite Key: A key made up of two or more attributes that together uniquely identify a record in a table.
  + Primary Key: A special candidate key selected to uniquely identify database records.
  + Alternate Key: A candidate key not chosen as the primary key.
  + Foreign Key: An attribute in one table that links to the primary key in another table.
  + Non-Key Attribute: Columns in a database table that do not serve as primary or candidate keys.
* Ideas/Techniques with Definition:
  + Unique Identification: The concept of ensuring that each record in a database can be distinctly identified from all others.
  + Database Design: The process of defining the structure of a database, including its tables, keys, and relationships.
* Keywords: relational database model, entities, relations, keys, primary key, foreign key, composite key, candidate key, alternate key, sports competition database
* Mistakes or Misleading Information:
  + The mention of "multi-value attributes" being avoided in relational databases is somewhat misleading without context. While true that relational databases typically normalize data to avoid multi-value attributes in a single column, modern databases can handle such data types (e.g., arrays in SQL) under specific use cases.
  + The example given for a composite key (staff name and staff title) could potentially be misleading without clarifying that both attributes together must uniquely identify each row. It's essential to ensure that no two rows have the same combination of these attributes, which might not always be guaranteed in practical scenarios depending on the dataset.
  + The explanation of keys is clear but lacks a direct connection to how these keys enforce relationships between tables, especially for beginners. A practical example showing how a foreign key in one table links to a primary key in another table to establish a relationship might enhance understanding.

Written by ChatGPT

[Knowledge check: Database structure | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/6eBa7/knowledge-check-database-structure)

* 5 of 5 basic questions but good test.

5 of 5 basic questions

[Module summary: Introduction to Databases | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/9g8HK/module-summary-introduction-to-databases) 3-24-2024 0450

Summary: This module provides a comprehensive introduction to databases, covering their basic concept, the structured query language (SQL) for database interaction, and the structure of databases including tables and key components. It prepares learners for working with databases by explaining the organization of data, the evolution of database technologies, and the practical applications of SQL. Acronyms:

* SQL: Structured Query Language
* DDL: Data Definition Language
* DML: Data Manipulation Language
* DQL: Data Query Language Terms:
* Database: A structured set of data held in a computer, especially one that is accessible in various ways.
* Data: Facts and statistics collected together for reference or analysis.
* Related Data: Data that is connected in some way, typically through shared attributes or relationships within a database.
* Database Applications: Software that utilizes databases to store, retrieve, and manipulate data for various applications.
* Operating Systems: Software that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. Definitions:
* Conceptual Level: A high-level overview of a system or idea without going into detailed implementation.
* Real World Examples: Instances or cases drawn from actual experience in society or industry that illustrate a particular theory or concept.
* Organize: To arrange or order things in a structured manner.
* Importance: The state or fact of being of great significance or value.
* Trends: A general direction in which something is developing or changing.
* Evolution: The gradual development of something, especially from a simple to a more complex form.
* Low Entry Level: Refers to something that is accessible with minimal prior knowledge or skill.
* Portability: The ease with which a software can be transferred from one computing environment to another. Technical Words:
* SQL Syntax: The set of rules that defines the combinations of symbols that are considered to be correctly structured statements or expressions in SQL.
* Commands: Instructions to a computer or software to perform a specific task.
* Database Table: A structure that organizes data into rows and columns.
* Columns: Vertical divisions of data in a table, representing a specific attribute.
* Rows: Horizontal divisions of data in a table, representing a single record.
* Data Types: Categories of data that tell the database what kind of value a particular column can hold.
* Keys: Attributes or sets of attributes that uniquely identify records within a table. Rarely Used Words:
* Comprehensive: Thorough and complete; covering or involving all aspects.
* Module: A unit of instruction or learning.
* Recap: To review or summarize the main points of something. Software Packages: Not mentioned in the transcript. Commands: Not explicitly mentioned in the transcript. Modules: Not mentioned in the transcript. Ideas:
* Database Management: The process of storing, retrieving, and managing data in databases.
* Data Organization: The practice of structuring and arranging data in a database for efficient access and management. Techniques: Not mentioned in the transcript. Important People: Not mentioned in the transcript. Important Dates: Not mentioned in the transcript. Theories: Not mentioned in the transcript. Keywords: ["database", "data", "SQL", "Structured Query Language", "database table", "columns", "rows", "data types", "keys", "DDL", "DML", "DQL"]

Written by Gemini, ChatGPT

[Module quiz: Introduction to Databases | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/exam/kQuwT/module-quiz-introduction-to-databases) 3-24-2024 0515

* Missed 2 partial questions. No wrong answers. No need to redo, but reviewed.

Week 2

[Numeric data types | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/gp5gK/numeric-data-types)

**Summary:** This video explains how databases use data types to ensure data is stored correctly. It focuses on numeric data types, specifically the difference between integers (whole numbers) and decimals (numbers with fractions).

**Detailed Breakdown:**

* **Acronyms**
  + DBMS (Database Management System)
* **Technical Words**
  + Data type: A classification for the kind of data that a database field can hold.
  + Column: A vertical division of a database table that holds a specific type of data.
  + Row: A horizontal division of a database table representing a single record.
  + Integer: A number without fractions or decimals (a whole number).
  + Decimal: A number that includes a decimal point and fractional parts
* **Rarely Used Words**
  + Rendered: Converted or displayed
* **Software Packages**
  + MySQL: A popular open-source database management system
* **Ideas/Techniques**
  + Using data types to enforce data integrity in a database
* **Keywords:** database, data type, numeric, integer, decimal, MySQL
* **Mistakes or Misleading Information**
  + **Minor point:** The video states that fractional numbers inserted into an integer column are automatically rounded up or down. Depending on the database system, they might be truncated (the decimal portion is simply cut off).

**Written by Gemini**

[Exercise: Working with numbers | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/hZ35y/exercise-working-with-numbers)<Lab> 3-24-2024 -0552

* Simple lab exercise. This is my second pass through the course so my lab exercise is still in the virtual lab. I did the optional part this round.

[Self review: Working with numbers | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/6ujUm/self-review-working-with-numbers) 3-24-2024 0609

* Basic test

[String data types | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/yh9SP/string-data-types)

**Summary:** This video explains string data types for storing text data in a database. It focuses on the difference between CHAR (fixed-length) and VARCHAR (variable-length) strings.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + String: A sequence of characters, used to represent text.
  + Alphanumeric: Characters that include both letters and numbers.
* **Commands/Modules**
  + CHAR(): A SQL command to define a fixed-length string column.
  + VARCHAR(): A SQL command to define a variable-length string column.
* **Mistakes or Misleading Information**
  + **Imprecise statement:** The video says "you should use string datatype when you intend to store data that contains a mix of character types". It would be more accurate to say that string data types are suitable for storing text that may contain a mix of letters, numbers, and symbols.

**Written by Gemini**

[Exercise: Working with strings | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/GH1e0/exercise-working-with-strings) 3-24-2024 0640

* Simple lab. I walked through my previous work.

[Self review: Working with strings | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/gVVB0/self-review-working-with-strings) 3-2024 0706

* Simple quiz, but I missed one. I thought TEXT was a trick and forgot it was a real data type. I (re)learned it is used for large amounts of text and CHAR is for under 255.

[Working with default values  | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/KZTUS/working-with-default-values)

**Summary:** This video introduces database constraints, which enforce data quality. It specifically focuses on NOT NULL (preventing empty fields) and DEFAULT (providing default values if no data is explicitly given).

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Constraint: A rule that limits what data can go into database fields.
  + Column: A vertical division of a database table holding a specific type of data.
  + Table: A structured way to organize data within a database.
  + Data type: Classification of the kind of data a field can hold (e.g., text, number).
* **Commands/Modules**
  + NOT NULL: SQL keyword to prevent a column from having empty values.
  + DEFAULT: SQL keyword to set a default value for a column.
  + CREATE TABLE: SQL command for creating a new database table.
* **Ideas/Techniques**
  + Using constraints to maintain data integrity
* **Keywords:** database, constraint, SQL, NOT NULL, DEFAULT, column, table, data type
* **Mistakes or Misleading Information**
  + **Minor point:** The phrase "prevents actions that would destroy links between tables" is often associated with FOREIGN KEY constraints, not NOT NULL or DEFAULT specifically.

**Written by Gemini**

[Self review: Working with default values | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/TEhqE/self-review-working-with-default-values)

I ended my last pass through this course with 80% on this test telling me I had problems since you get 3 tries. Lets see what happens this pass. I missed one. I most likely missed the same question last time. Below is the question and why I’m sticking with it.  
A close-up of a text

Description automatically generated

Im sticking with my answer, but to pass the test I will change to True.

False.

Here's why: From Gemini

Database Constraints (Broader Concept): Database constraints indeed limit the kind of data that can be stored in a table, ensuring accuracy and consistency.

DEFAULT Constraints (Specific Type): DEFAULT constraints do contribute to this, but their primary role is to provide a default value if a user doesn't explicitly provide one during data insertion.

Other Constraints Limiting Data Values:

NOT NULL: Prevents empty fields, enforcing that a value must be provided.

CHECK: Limits the range of acceptable values based on a condition (e.g., age must be greater than 18).

UNIQUE: Ensures that values in a column are distinct (no duplicates).

False. Frome ChatGPT

Database default constraints do not limit the values that can be stored in a table. Instead, they specify a default value for a column when no value is specified during the insertion of a new record. If a value is provided for the column during the insert operation, the default constraint is overridden by the provided value. Default constraints are used to ensure that a column always has a value, preventing null values unless explicitly allowed, but they do not restrict or limit the range of values that can be entered into a column.

[Choosing the right data type for a column  | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/ZuWlh/choosing-the-right-data-type-for-a-column)<Lab> 3-24-2024 0800

* These labs are extremely basic, but for muscle memory I am going through the exercises again. It’s a good thing as I do not use SQL commands often and often get them wrong.

[Additional resources | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/bTAfL/additional-resources) 3-24-2024 1300

# Additional resources

The following list of resources explore the meaning and role of different data types in databases. They also provide good examples for how to declare and use the data types in SQL.

* [W3schools](https://www.w3schools.com/sql/sql_datatypes.asp)
* [W3resource](https://www.w3resource.com/mysql/mysql-data-types.php)
* [LearnSQL](https://learnsql.com/blog/understanding-numerical-data-types-sql/)
* [Microsoft](https://docs.microsoft.com/en-us/sql/t-sql/data-types/decimal-and-numeric-transact-sql?view=sql-server-ver16s)
* [MySQL](https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html)

[CREATE and DROP database | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/7mLKU/create-and-drop-database) 3-24-2024 1330

[ALTER TABLE statement | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/Jvgtv/alter-table-statement) 3-24-2024 1345

**Short Summary:** This video explains how to use SQL to create tables within a database, providing a foundation for organizing data. It specifically covers the CREATE TABLE statement and the importance of having a pre-existing database.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column: A vertical division of a table that holds a specific type of data.
  + Datatype: The classification of the kind of data a column can hold (e.g., text, numbers, dates).
  + Syntax: The specific arrangement of words and symbols in a programming language.
* **Rarely Used Words**
  + Respective: Corresponding to something previously mentioned.
  + Execute: Carry out or put into effect (e.g., executing a SQL statement).
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (e.g., MySQL, PostgreSQL).
* **Commands/Modules**
  + CREATE TABLE: The SQL command used to create a new table within a database.
* **Ideas/Techniques**
  + Data organization: The process of structuring data for easy retrieval and analysis.
* **Keywords:** database, table, column, datatype, SQL, CREATE TABLE

**Mistakes or Misleading Information**

* **Minor Oversimplification:** While VARCHAR is versatile, there are more specific text datatypes (like TEXT) for larger amounts of text.

**Written by Gemini**

[ALTER TABLE statement | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/Jvgtv/alter-table-statement) 3-24-2024 1430

**Short Summary:** This video teaches how to modify existing database tables using SQL. It covers adding, deleting, and altering columns within a table using the ALTER TABLE statement.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column: A vertical division of a table that holds a specific type of data.
  + Datatype: The classification of the kind of data a column can hold (e.g., text, numbers, dates).
  + Syntax: The specific arrangement of words and symbols in a programming language.
* **Rarely Used Words**
  + Static: Unchanging.
  + Restructuring: Changing the structure of something.
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (implied, but not explicitly named).
* **Commands/Modules**
  + ALTER TABLE: The SQL command used to modify an existing table's structure.
  + ADD: Used within ALTER TABLE to add new columns.
  + DROP COLUMN: Used within ALTER TABLE to remove columns.
  + MODIFY: Used within ALTER TABLE to change column attributes (like datatype).
* **Ideas/Techniques**
  + Database Modification: The process of making changes to the structure of a database table.
* **Keywords:** database, SQL, table, column, datatype, ALTER TABLE, ADD, DROP COLUMN, MODIFY

**Mistakes or Misleading Information**

* **Confirmation:** Modern database systems usually ask for confirmation before dropping columns to prevent accidental data loss. The video makes it sound like this is an optional step.
* **Oversimplifies VARCHAR:** While VARCHAR is versatile, databases often have a dedicated TEXT datatype for larger text fields offering better performance.
* **Im running through the exercises on MySQL on my desktio.**

**Written by Gemini**

A screenshot of a computer screen

Description automatically generated

[INSERT statement | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/iziJa/insert-statement) 3-24-2024 1545

**Short Summary:** This video teaches how to add new rows of data to existing database tables using SQL's INSERT INTO statement. It covers the basic syntax, inserting multiple rows, and the importance of matching values to the correct columns.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column: A vertical division of a table that holds a specific type of data.
  + Datatype: The classification of the kind of data a column can hold (e.g., text, numbers, dates).
  + Syntax: The specific arrangement of words and symbols in a programming language.
  + Clause: A distinct part of an SQL statement with a specific purpose.
* **Rarely Used Words**
  + Execute: Carry out or put into effect (e.g., executing a SQL statement).
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (implied, but not explicitly named).
* **Commands/Modules**
  + INSERT INTO: The SQL command used to add new rows to a table.
  + VALUES: Keyword within INSERT INTO to specify the data being inserted.
  + SELECT: The SQL command used to retrieve data from a table.
  + FROM: Keyword within SELECT to indicate the table to query.
* **Ideas/Techniques**
  + Data Insertion: The process of adding new rows of information to a database table.
* **Keywords:** SQL, database, table, column, INSERT INTO, VALUES, SELECT, FROM

**Mistakes or Misleading Information**

* **Slight Imprecision:** While the SELECT \* syntax is shown to view data, it's often recommended in practice to explicitly list the columns you want for better clarity and performance.
  + Example: SELECT ID, Name, Age, Start\_date FROM players

**Written by Gemini**

[Creating tables | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/IW5IZ/creating-tables) 3-24-2024 1600

This reading covers things the create table commands we have already covered.

**Short Summary:** This reading explains how to create database tables using the SQL CREATE TABLE statement. It emphasizes best practices like choosing meaningful names, appropriate datatypes, and considering storage requirements.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
  + DDL (Data Definition Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column (or Field): A vertical division of a table that holds a specific type of data.
  + Datatype: The classification of the kind of data a column can hold (e.g., text, numbers, dates).
* **Rarely Used Words**
  + Subset: A smaller set contained within a larger set.
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (e.g., MySQL, SQL Server, Oracle).
* **Commands/Modules**
  + CREATE TABLE: The SQL command used to create a new table in a database.
  + INT: A common datatype used to store integer numbers.
  + VARCHAR: A common datatype used to store variable-length text strings.
* **Ideas/Techniques**
  + Database Table Design: The process of planning the structure of a database table.
* **Keywords:** SQL, CREATE TABLE, database, table, column, datatype, INT, VARCHAR

**Mistakes or Misleading Information**

* **Minor Oversimplification:** While INT is common for integers, some databases might offer more specialized integer types (like SMALLINT for smaller ranges) that can save storage space.
* **Storage Imprecision:** The VARCHAR storage explanation needs slight adjustment. Some databases might have overhead beyond the 2 bytes for length information.

**Written by Gemini**

[Exercise: Create Database, create table and insert data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/uHsCL/exercise-create-database-create-table-and-insert-data) 3-24-2024 1636

Summary: This lesson provides a comprehensive guide on creating a database, a table within that database, and inserting data into the table using MySQL, aimed at assisting Mr. John Ericson in digitizing his bookshop's customer records. It covers the steps from initiating a database called "bookshop" to populating it with customer information through structured SQL commands.

* **Acronyms**
  + SQL: Structured Query Language, used for managing and manipulating databases.
* **Technical Words with Definition**
  + Database: A structured set of data held in a computer, especially one that is accessible in various ways.
  + Table: In the context of databases, a table is a collection of related data held in a structured format within a database; it consists of columns and rows.
  + Data Type: A classification identifying one of various types of data, such as integer, floating-point, or string, that determines the possible values for that type, the operations that can be done on that type, and the way values of that type are stored.
* **Commands/Modules with Definition**
  + **CREATE DATABASE**: A SQL command used to create a new database.
  + **USE**: A SQL command used to select a specific database to work with.
  + **CREATE TABLE**: A SQL command used to create a new table in the database.
  + **INSERT INTO**: A SQL command used to insert new data into a table.
  + **SELECT \* FROM**: A SQL command used to select and display all records from a table.
* **Ideas/Techniques with Definition**
  + Database Initialization: The process of creating a new database to store data.
  + Table Definition: Defining a table's structure in a database by specifying its columns and data types.
  + Data Insertion: The process of adding new records to a database table.
* **Mistakes or Misleading Information**
  + The lesson is clear and accurately presents the steps to create a database, table, and insert data using SQL commands. No misleading information or mistakes are noted.

Python list of Keywords: **keywords = ['CREATE DATABASE','USE','CREATE TABLE','INSERT INTO','SELECT \* FROM','Database','Table','Data Type','Database Initialization','Table Definition','Data Insertion']**

*Written by ChatGPT*

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

[SELECT statement | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/bvrDY/select-statement) 3-24-2024 1700

**Short Summary:** This video teaches how to use the SQL SELECT statement to retrieve data from database tables. It covers selecting specific columns, selecting all columns (using '\*'), and the basic syntax of the statement.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column: A vertical division of a table that holds a specific type of data.
  + Query: A request for information from a database.
  + Syntax: The specific arrangement of words and symbols in a programming language.
* **Rarely Used Words**
  + Concatenation: The process of joining things (like text) end-to-end.
  + Shorthand: A shortened way of writing something.
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (implied, but not explicitly named).
* **Commands/Modules**
  + SELECT: The SQL command used to retrieve data from a table.
  + FROM: Keyword within SELECT to specify the table to query.
  + Asterisk (\*): A symbol used as a wildcard in SELECT to represent all columns.
* **Ideas/Techniques**
  + Data Retrieval: The process of getting information from a database.
* **Keywords:** SQL, SELECT, FROM, database, table, column, query

**Mistakes or Misleading Information**

* **Imprecise Term:** While 'math calculations' is mentioned, SQL itself usually provides functions for calculations, not the core SELECT statement. Example: SELECT price \* quantity AS total\_cost FROM sales\_data

**Written by Gemini**

[INSERT INTO SELECT statement | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/PGdFO/insert-into-select-statement) 3-24-2024 1710

**Short Summary:** This video explains how to use the SQL INSERT INTO SELECT statement to copy data from columns in one table (source) into a different table (target). It covers the basic syntax and provides an example with a soccer club database.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: A structured collection of data stored electronically.
  + Table: A way to organize data within a database, consisting of rows and columns.
  + Column: A vertical division of a table that holds a specific type of data.
  + Query: A request for information from a database.
  + Syntax: The specific arrangement of words and symbols in a programming language.
  + Source Table: The table from which data is being copied.
  + Target Table: The table where data is being inserted.
* **Rarely Used Words**
  + Populate: To fill with information.
* **Software Packages**
  + Database Management System (DBMS): Software that manages databases (implied, but not explicitly named).
* **Commands/Modules**
  + INSERT INTO: The SQL command used to insert new rows into a table.
  + SELECT: The SQL command used to retrieve data from a table.
  + FROM: Keyword within SQL statements to specify the table to use.
* **Ideas/Techniques**
  + Data Transfer: The process of moving data from one table to another.
* keywords = ["SQL", "INSERT", "INTO", "SELECT", "FROM", "database", "table", "column", "query"]

**Mistakes or Misleading Information**

* **Slight Imprecision:** The video suggests that data must be arranged in a specific order within the source table. While this might simplify the example, INSERT INTO SELECT can be used with more complex queries to rearrange data during the transfer if needed.

**Written by Gemini**

[Exercise: Practicing table creation | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/wjaYf/exercise-practicing-table-creation) 3-24-2024 1709

* Nothing

[Knowledge check: Create, insert and select | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/cd2k7/knowledge-check-create-insert-and-select) 3-24-2024 1850

* Easy quiz 5 of 5

[Updating data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/h0BDJ/updating-data)

**Summary:** This video tutorial explains how to use the SQL UPDATE statement to modify data in a database table. It demonstrates updating a single student's record and then shows how to update multiple records at once based on specific conditions.

**Detailed Breakdown:**

* **Acronyms**
  + SQL (Structured Query Language)
* **Technical Words**
  + Database: An organized collection of data, typically stored electronically.
  + Table: A structured set of data arranged in rows and columns within a database.
  + Column: A vertical division in a table representing a specific category of data.
  + Row/Record: A horizontal division in a table representing a single entry.
  + Syntax: The specific rules governing the structure of a programming language.
* **Rarely Used Words**
  + Clause: A distinct part of a statement or sentence with its own subject and verb.
  + Syntax: Arrangement of words in a sentence to show their relationship with each other
* **Software Packages**
  + phpMyAdmin: A web-based tool for administering MySQL databases.
* **Commands/Modules**
  + UPDATE: SQL command used to modify existing data in a database table.
  + SET: SQL clause used to specify the columns and new values for an update.
  + WHERE: SQL clause used to filter the rows to be updated.
* **Ideas/Techniques**
  + Data Modification: The process of changing existing data within a database.
  + Conditional Updates: Targeting specific rows for updates based on defined criteria.

**Mistakes or Misleading Information**

* The tutorial focuses narrowly on the UPDATE statement and doesn't mention best practices for data integrity, like backups or testing updates in a staging environment before applying them to a production database.
* keywords = ['UPDATE', 'SET', 'WHERE', 'student', 'home address', 'contact number', 'department', 'college address', 'engineering']

**Written by Gemini**

[Deleting data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/nyuEq/deleting-data) 3-25-2024 0415

Short Summary: This video demonstrates how to delete records in a database using SQL. It covers deleting single records, multiple records based on criteria, and deleting all records from a table within the phpMyAdmin interface.

Detailed Breakdown:

* Acronyms
  + SQL (Structured Query Language)
* Technical words with definition
  + Database: An organized collection of data managed by a database management system.
  + Table: A structure within a database to store related data in rows and columns.
  + Record: A single row in a database table, representing an individual data item.
  + Column: A vertical division of a table representing a specific field of data.
  + Query: An instruction sent to a database, often for retrieving or modifying data.
* Rarely used words for a 12-year-old with definition
  + Instance: A specific occurrence or example of something (in this context, a record within a table).
* Software packages with definition
  + phpMyAdmin: A web-based tool for administering MySQL and MariaDB databases.
* Commands/modules with definition and sample usage
  + DELETE: An SQL command to remove records from a table. Ex: DELETE FROM customers WHERE customerID = 5
  + FROM: An SQL keyword specifying the table from which to delete records.
  + WHERE: An SQL keyword to introduce conditions for filtering the records to delete.
* Ideas/techniques with definition
  + Conditional Deletion: Deleting specific records based on conditions defined in the WHERE clause of an SQL statement.
* Keywords
  + keywords = ['DELETE', 'FROM', 'WHERE', 'SELECT']

Mistakes or Misleading Information:

* The video suggests that running a DELETE FROM student\_table command without a WHERE clause would delete all records. While technically correct, **emphasizing the risks of such actions is crucial.** It's essential to teach responsible SQL practices and the importance of backups.

Written by Gemini

[Exercise: Record deletion | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/M0xh5/exercise-record-deletion) 3-25-2024 0500

* The exercise asks us to copy this but you get an error. You do not use quotes for the table name.
* <BAD> INSERT INTO 'customers' ('customerID', 'customerName', 'customerAddress')
* <Good> INSERT INTO customers ('customerID', 'customerName', 'customerAddress')

[Self-review: Record deletion | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/hlc0w/self-review-record-deletion/attempt?redirectToCover=true) 3-25-2024 0504

5 of 5 basic stuff

[Knowledge check: Update and Delete | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/cjRXt/knowledge-check-update-and-delete) 3-25-2024 0509

* A little harder than the previous test, but still basic. 5 of 5

[Module summary: Create, Read, Update and Delete (CRUD) Operations | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/OOyHj/module-summary-create-read-update-and-delete-crud-operations)

Summary: This module provides a comprehensive overview of CRUD operations in SQL, including creating, reading, updating, and deleting data within a database. It covers SQL data types, constraints, and essential SQL commands, equipping learners with the skills to manipulate and manage data effectively.

### Acronyms

* CRUD: Create, Read, Update, Delete
* SQL: Structured Query Language

### Technical Words with Definition

* **Database**: A structured set of data held in a computer, especially one that is accessible in various ways.
* **SQL Data Types**: Categories of data that can be stored in SQL databases, such as numeric and string types.
* **Constraints**: Rules applied to data fields in a database to ensure the integrity and accuracy of the data.

### Software Packages with Definition

* **SQL Database Management Systems**: Software applications used to manage SQL databases, like MySQL, PostgreSQL, and SQLite.

### Commands/Modules with Definition

* **SELECT**: Retrieves data from a database. Example: **SELECT \* FROM users**
* **INSERT INTO**: Adds new data to a table. Example: **INSERT INTO users (name, age) VALUES ('John', 30)**
* **UPDATE**: Modifies existing data in a table. Example: **UPDATE users SET age = 31 WHERE name = 'John'**
* **DELETE**: Removes data from a table. Example: **DELETE FROM users WHERE name = 'John'**
* **VARCHAR**: A variable-length string data type.

### Ideas/Techniques with Definition

* **Numeric Data Types**: Types of data that represent numbers, including integers and decimals.
* **String Data Types**: Types of data that represent text, including CHAR and VARCHAR.

### Mistakes or Misleading Information

* There might be a misunderstanding in the summary about the use of "uncertain" which likely was intended to be "insert." To clarify: "Uncertain" is not related to SQL operations; the correct term is "insert" as in the SQL **INSERT INTO** command used to add data to tables.

keywords = ['CRUD', 'SQL', 'Database', 'Data Types', 'Constraints', 'Decimal', 'VARCHAR', 'SELECT', 'INSERT INTO', 'UPDATE', 'DELETE']

Written by ChatGPT

[Module quiz: Create, Read, Update and Delete (CRUD) Operations | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/exam/9EHu4/module-quiz-create-read-update-and-delete-crud-operations) 3-24-2024 0539

* Missed a partial question. Second pass was 100%.

Week 3

[SQL Arithmetic Operators | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/z8SNL/sql-arithmetic-operators) 3-25-2024 0559

Summary: This video introduces SQL arithmetic operators, explaining their importance in querying and manipulating data within a database. It covers the basics of operators, their types, and provides examples on how to perform arithmetic operations like addition, subtraction, multiplication, division, and modulus using SQL.

### Acronyms

* SQL: Structured Query Language

### Technical Words with Definition

* **Operators**: Symbols or words that perform operations on operands in SQL.
* **Operands**: Values on which SQL operators perform operations.

### Rarely Used Words with Definition

* **Modulus**: An arithmetic operation that finds the remainder of the division of one number by another.

### Commands/Modules with Definition

* **SELECT**: A SQL command used to select data from a database.

keywords = ['SQL', 'Operators', 'Operands', 'Arithmetic Operators', 'SELECT', 'Modulus', 'Addition', 'Subtraction', 'Multiplication', 'Division']

Written by ChatGPT

[SQL Arithmetic Operator Examples | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/FnizP/sql-arithmetic-operator-examples) 3-24-2024 0610

Summary: This reading segment educates on the application of SQL arithmetic operators, including addition, subtraction, multiplication, division, and modulus, within SELECT and WHERE clauses for effective data manipulation and querying in SQL databases.

### Acronyms

* SQL: Structured Query Language

### Technical Words with Definition

* **Arithmetic Operators**: Symbols used in mathematics and programming to perform operations such as addition (+), subtraction (-), multiplication (\*), division (/), and modulus (%).
* **Operand**: A value that an operator acts on. In SQL, operands are typically column values.

### Rarely Used Words with Definition

* **Modulus Operator (%):** Provides the remainder from a division operation.

### Commands/Modules with Definition

* **SELECT**: Retrieves data from a database. Example: **SELECT salary + allowance FROM employee;**
* **WHERE**: Filters records that fulfill a specified condition. Example: **SELECT \* FROM employee WHERE salary - tax = 50000;**

### Ideas/Techniques with Definition

* **Using Arithmetic Operators in SQL**: Incorporating mathematical operations in SQL queries to manipulate and analyze numerical data stored in database tables.

keywords = ['SELECT', 'FROM', 'WHERE', '+', '-', '\*', '/', '%', 'salary', 'allowance', 'tax', 'hours']

Written by ChatGPT

[Operators in use | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/UV2ym/operators-in-use) 3-25-2024 0618

Summary: This video tutorial demonstrates practical applications of SQL arithmetic operators in managing employee data within a database. It showcases how to perform operations such as adding bonuses, deducting amounts, doubling salaries, calculating monthly salaries, and determining if employee IDs are even or odd using SQL commands.

### Acronyms

* SQL: Structured Query Language

### Rarely Used Words with Definition

* **Modulus Operator**: A mathematical operation that returns the remainder of a division.

### Commands/Modules with Definition

* **SELECT**: Retrieves specified columns from one or more tables in a SQL database. Example: **SELECT salary + 500 FROM employee;**

### Ideas/Techniques with Definition

* **Data Manipulation**: The process of modifying data to make it organized or easier to read.

keywords = ['SELECT', 'modulus']

Written by ChatGPT

[SQL Comparison operators | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/eGlgL/sql-comparison-operators) 3-25-2024 0625

Summary: This video tutorial delves into the use of SQL comparison operators for data filtration and manipulation, showcasing practical applications within a company's employee database. It explains how to use operators like equal to, less than, greater than, less than or equal to, greater than or equal to, and not equal to, for various data retrieval needs.

### Acronyms

* SQL: Structured Query Language

### Technical Words with Definition

* **Comparison Operators**: Symbols used in SQL to compare two values or expressions, resulting in a boolean outcome (true or false).

### Commands/Modules with Definition

* **SELECT**: Command used to retrieve data from a database. Example: **SELECT \* FROM employee;**
* **WHERE**: Clause used in SQL to specify criteria for filtering data. Example: **SELECT \* FROM employee WHERE salary = 18000;**

### Ideas/Techniques with Definition

* **Data Extraction**: The process of retrieving specific data from a database based on certain criteria, often using comparison operators for filtering.

### Keywords

keywords = ['SELECT', 'FROM', 'WHERE']

Written by ChatGPT

[SQL Comparison operator examples | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/vR03i/sql-comparison-operator-examples)

[SQL Comparison operator examples | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/vR03i/sql-comparison-operator-examples)

Summary: This reading elaborates on SQL comparison operators, illustrating their application in various scenarios within the WHERE clause of SELECT statements to filter data. It refreshes knowledge on operators such as equality, inequality, greater than, and less than, providing examples for a better understanding of how to utilize them in SQL queries. Acronyms: No acronyms mentioned in the transcript. Terms:

* SQL Comparison Operators: Operators used in SQL to compare two expressions in a query.
* WHERE Clause: A part of SQL statement that filters records to fetch only those that meet a certain condition.
* SELECT Statement: A SQL command used to select data from a database.
* Records: Rows in a database table, representing individual entries or data points.
* Query: A request for data or information from a database.
* Expression: A combination of symbols and operators that the database evaluates to produce a result. Technical Words:
* Operator: A symbol or function that indicates an operation to be performed, e.g., addition, multiplication, comparison.
* Equality ( = ): An operator that tests whether two expressions are equal.
* Inequality ( <> or != ): An operator that tests whether two expressions are not equal.
* Greater Than ( > ): An operator that tests if the left-hand expression is greater than the right-hand expression.
* Greater Than or Equal ( >= ): An operator that tests if the left-hand expression is greater than or equal to the right-hand expression.
* Less Than ( < ): An operator that tests if the left-hand expression is less than the right-hand expression.
* Less Than or Equal ( <= ): An operator that tests if the left-hand expression is less than or equal to the right-hand expression. Rarely Used Words:
* Literal: A notation for representing a fixed value in source code. Software Packages: Not mentioned in the transcript. Commands:
* SELECT \* FROM [table] WHERE [condition];: A basic SQL command structure used to retrieve data from a specified table where the data matches a specific condition. Modules: Not mentioned in the transcript. Ideas:
* Filtering Data: The process of specifying conditions in a database query to limit the records returned.
* Data Retrieval: The process of extracting data from a database in response to a query. Techniques: Not mentioned in the transcript. Important People: Not mentioned in the transcript. Important Dates: Not mentioned in the transcript. Theories: Not mentioned in the transcript. Keywords: ["SQL Comparison Operators", "WHERE Clause", "SELECT Statement", "Records", "Query", "Equality", "Inequality", "Greater Than", "Less Than", "Operator", "Expression"]

Notes taken by ChatGPT

[Knowledge Check: Operators | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/E1Snu/knowledge-check-operators)

* Very simple test, but the fun was needing to reinstall MySQL because I forgot password. First the LLM tried to get me to do the reset password, but that was a fail. The my.ini file was nowhere to be found and even when I put a new one in the directory it said I should the program never found it. I eventually uninstalled and reinstalled. But the uninstall process had issues with removing everything so I had to manually do it. But most likely the difficulties were all due to my newbiness.
* I created a DB and tables needed by using these commands
* CREATE DATABASE IF NOT EXISTS CourseraDB;
* USE CourseraDB;
* CREATE TABLE IF NOT EXISTS invoices (InvoiceID INT AUTO\_INCREMENT PRIMARY KEY,Total DECIMAL(10,2) NOT NULL);
* INSERT INTO invoices (Total) VALUES (1.98);
* INSERT INTO invoices (Total) VALUES (3.96);
* INSERT INTO invoices (Total) VALUES (5.94);
* INSERT INTO invoices (Total) VALUES (8.91);
* INSERT INTO invoices (Total) VALUES (13.86);
* INSERT INTO invoices (Total) VALUES (0.99);

[ORDER BY clause | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/ZIdK6/order-by-clause) 3-25-2024 1300

Summary: This text explains the purpose and usage of the SQL ORDER BY clause, demonstrating how it sorts data in a database table based on specified column(s) in either ascending or descending order.

Acronyms:

* ASC: Ascending Order
* DESC: Descending Order
* SQL: Structured Query Language

Terms:

* ORDER BY Clause: A clause used in SQL to sort the data returned by a SELECT statement in ascending or descending order.
* Select Statement: A SQL command used to select data from a database.
* Column: A vertical entity in a table that contains all information associated with a specific field.
* Table: A collection of related data held in a table format within a database.
* ID: An abbreviation for 'Identification', often used as a column name for a unique identifier in a database.

Technical Words:

* Clause: A part of a SQL statement that performs a specific function within the statement.
* Syntax: The set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in SQL.
* Keyword: Reserved words in SQL used to perform specific functions.

Commands:

* **SELECT**: Retrieves data from a database**. SELECT name, salary FROM employees**;
* **ORDER BY**: Specifies the order in which the SELECT statement's results should be returned. **SELECT \* FROM products ORDER BY price DESC;**

Ideas:

* Sorting Data: The process of ordering data in a specific sequence, either ascending or descending.
* Alphabetical Order: A sequence that is arranged in the same order as the alphabet, A-Z.

Techniques:

* Multiple Column Sorting: Ordering data by more than one column, where the first column is sorted, and within each set of identical values for the first column, the second column is sorted, and so on.

Keywords: ["ORDER BY", "SELECT", "ID", "ASC", "DESC", "SQL"]

Notes taken by ChatGPT

Top of Form

[Types of ordering / sorting | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/qmCQQ/types-of-ordering-sorting) 3-25-2024 1410

* Super basic stuff, but good review if you need it.

[WHERE clause | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/bxBvM/where-clause) 3-25-2024 1439

Summary: This video explains the use of the SQL WHERE clause for filtering data in database tables, demonstrating how to apply conditions to select statements and the use of various operators to refine queries.

Acronyms:

* SQL: Structured Query Language

Terms:

* WHERE Clause: A clause used in SQL queries to filter results based on specified conditions.
* SELECT Statement: A SQL command used to retrieve data from a database.
* FROM Clause: A part of the SELECT statement in SQL that specifies the table to retrieve the data from.
* Condition: A statement in SQL that must be true for a record to be included in the query results.

Technical Words:

* Operators: Symbols or keywords in SQL that specify the type of comparison to be made.
* ASC: Ascending order, used in ORDER BY clause to sort data from smallest to largest or alphabetically A-Z.
* DESC: Descending order, used in ORDER BY clause to sort data from largest to smallest or alphabetically Z-A.
* BETWEEN Operator: Used to filter data within a certain range.
* LIKE Operator: Used for pattern matching in SQL.
* IN Operator: Used to filter data against a list of values.
* Operand: A value or column reference on which the operator acts to produce a result.

Rarely used words:

* Operand: In the context of SQL, this term refers to the specific values or columns to which an operator is applied.
* Wild Card Characters: Special characters in SQL, like **%** and **\_**, used with the LIKE operator to represent one or more characters in a pattern match.

Commands:

* **SELECT:** Retrieves data from a database. Example**: SELECT first\_name, last\_name FROM students; retrieves all first and last names from the 'students' table.**
* **FROM:** Specifies the table to select the data from. Example**: SELECT \* FROM employees; retrieves all data from the 'employees' table.**
* **WHERE:** Filters the result set to include only rows that meet specified conditions**. Example: SELECT \* FROM orders WHERE order\_date >= '2023-01-01';**
* **ORDER BY:** Arranges the result set of a query by one or more columns. Example**: SELECT name, age FROM users ORDER BY age DESC;**

Keywords: ["SQL", "WHERE clause", "SELECT statement", "FROM clause", "ORDER BY clause", "Condition", "Operators", "ASC", "DESC", "BETWEEN", "LIKE", "IN", "Operand"]

Notes taken by ChatGPTTop of Form

[WHERE Clause uses | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/gPNOw/where-clause-uses)3-25-2024 1430

* Very basic but good review.

[ORDER BY and WHERE | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/fE4Zl/order-by-and-where) 3-25-2024 1440

* Did exercises in my MySQL database

[SELECT DISTINCT clause | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/AOigL/select-distinct-clause) 3-25-2024 1455

**Short Summary:** This video explains how to use the SQL SELECT DISTINCT statement to remove duplicate values from query results. It covers examples with single columns, multiple columns, and how SELECT DISTINCT handles null values.

**Acronyms**

* **SQL:** Structured Query Language

**Terms**

* **Database:** A structured collection of data stored electronically.
* **Table:** A structure within a database that holds data in rows and columns.
* **Column:** A vertical field in a table holding a specific type of data.
* **Row (Record):** A single horizontal entry in a table.

**Technical Words**

* **SELECT:** A SQL command for retrieving data from a database.
* **DISTINCT:** A SQL keyword to eliminate duplicates from results.
* **FROM:** A SQL keyword specifying the table(s) for data retrieval.
* **Query:** A request for data or information from a database.

**Commands**

* **SELECT DISTINCT:** Retrieves unique values from specified column(s). Example: **SELECT DISTINCT country FROM student;**

**Keywords:** SQL, SELECT, DISTINCT, database, table, column, row, duplicate, null, query

Notes taken by Gemini

[SELECT DISTINCT clause in use | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/wvmvF/select-distinct-clause-in-use) 3-25-2024 1510

* Basic stuff, but I went through the motions on my DB.

[Module summary: SQL operators and sorting and filtering data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/b1905/module-summary-sql-operators-and-sorting-and-filtering-data) 3-25-2024 1545

* Just a module summary

[Module quiz: SQL operators and sorting and filtering data | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/exam/ZMAJY/module-quiz-sql-operators-and-sorting-and-filtering-data) 3-26-2024 0439

Basic quiz 100%

[Database schema | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/BKEzz/database-schema) 3-36-2024 0453

Short Summary: This video explains the concept of a database schema. It discusses how the definition of a schema varies across database systems (MySQL, SQL Server, PostgreSQL, Oracle) and outlines the benefits of using schemas for data organization and security.

Detailed Breakdown:

* Acronyms
  + SQL (Structured Query Language)
* Technical words with definition
  + Schema: A structure or blueprint representing how data is organized within a database.
  + Table: A structure within a database storing data in rows and columns.
  + Column: A vertical division of a table representing a specific field of data.
  + Datatype: The classification of a particular type of data, determining possible values and operations (e.g., integer, text, date).
  + Key: A field or combination of fields used to uniquely identify a record within a table.
  + Relationship: A connection or association between tables in a database.
* Rarely used words for a 12-year-old with definition
  + Namespace: A container within which names (of database objects) must be unique to avoid conflicts.
* Software packages with definition
  + MySQL: A popular open-source relational database management system.
  + SQL Server: A relational database management system developed by Microsoft.
  + PostgreSQL: An object-relational database management system emphasizing extensibility and standards compliance.
  + Oracle: A commercial relational database management system known for scalability and enterprise features.
* Ideas/techniques with definition
  + Database Schema Design: The process of planning and defining the structure of a database, including tables, columns, relationships, and datatypes.
* Keywords
  + keywords = ['SELECT', 'FROM', 'WHERE', 'JOIN', 'GROUP BY', 'ORDER BY']

Mistakes or Misleading Information:

* **Conflating Schema and Database:** In MySQL, the terms are sometimes used interchangeably, but it's important to understand that a schema is a component within a database, not the entire database itself.
* **Oversimplifying Relationships:** The video mentions relationships but doesn't delve into the different types (one-to-one, one-to-many, many-to-many), which are crucial for schema design.

Written by Gemini

[Exploring database schema | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/gvj0H/exploring-database-schema)

Short Summary: This reading explains the concept of a database schema, including its importance and the three main types (conceptual, internal, and external). It emphasizes how a well-designed schema aids in organizing data, writing queries, and avoiding costly reverse-engineering.

Detailed Breakdown:

* Acronyms
  + ER-D (Entity Relationship Diagram)
* Technical words with definition
  + Schema: The structure or blueprint defining how data is organized within a database.
  + Entity: A real-world object or concept represented in a database (e.g., employee, department).
  + Attribute: A characteristic or property of an entity (e.g., employee name, department ID).
  + Relationship: An association between entities (e.g., an employee 'belongs to' a department).
* Rarely used words for a 12-year-old with definition
  + Reverse-engineering: Analyzing a system to understand its design and how it was built.
* Ideas/techniques with definition
  + Database Schema Design (Data Modeling): The process of planning and defining the structure of a database.
  + Three-Schema Architecture: A way of organizing database design into conceptual, internal, and external levels.
  + View Schema: A customized portion of the database tailored to a specific user or group of users.
* Keywords
  + keywords = ['SELECT', 'FROM', 'WHERE', 'JOIN', 'CREATE TABLE', 'ALTER TABLE']

Mistakes or Misleading Information:

* **Oversimplification of Physical Implementation:** The internal/physical schema description focuses on disk storage. In reality, database systems use complex indexing, caching, and optimization techniques that the text doesn't address.

Written by Gemini

[Schema in use | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/6TAf8/schema-in-use)

Short Summary: This video explains how to create a basic database schema using SQL. It demonstrates setting up a shopping cart database with three tables (customer, product, cart\_order), including the use of primary and foreign keys to establish relationships.

Detailed Breakdown:

* Acronyms
  + SQL (Structured Query Language)
* Technical words with definition
  + Database: An organized collection of data managed by a database management system.
  + Schema: The structure or blueprint defining how data is organized within a database.
  + Table: A structure within a database storing data in rows and columns.
  + Field/Column: A vertical division of a table representing a specific data attribute.
  + Datatype: Classifies the type of data a field can store (e.g., integer, text, date).
  + Primary Key: A field or combination of fields that uniquely identifies each record in a table.
  + Foreign Key: A field in a table that references a primary key in another table, establishing a relationship between them.
* Rarely Used Words for a 12-year-old
  + Parent/Referenced Table: The table containing the primary key that a foreign key points to.
* Software Packages
  + MySQL (implied): A popular relational database management system.
* Commands/Modules with definition
  + CREATE DATABASE database\_name: Creates a new database.
  + CREATE TABLE table\_name (...): Creates a new table with specified columns and datatypes.
  + PRIMARY KEY: Designates a column as the primary key.
  + FOREIGN KEY: Designates a column as a foreign key.
  + REFERENCES table\_name (column\_name): Establishes the reference between a foreign key and its target primary key.
* Ideas/techniques with definition
  + Database Schema Design: Planning and defining the structure of a database, including tables, columns, relationships, and keys.
* Keywords
  + keywords = ['SELECT', 'FROM', 'WHERE', 'CREATE DATABASE', 'CREATE TABLE', 'PRIMARY KEY', 'FOREIGN KEY', 'REFERENCES']

Mistakes or Misleading Information:

* **Omits 'USE' statement:** The video doesn't explicitly show the USE shopping\_cart\_db; command needed to work within the newly created database.
* **Conceptual Simplification**: Descrikes the balance between clarity and technical precision. A more advanced discussion would highlight that foreign keys enforce referential integrity for data consistency.

Written by Gemini

[Types of database schema | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/GEB9e/types-of-database-schema) 3-26-2024 0555

Short Summary: This video introduces logical and physical database schemas. It explains how logical schemas define data organization and relationships (using ER modeling), while physical schemas dictate how data is stored on disk (using SQL).

Detailed Breakdown:

* Acronyms
  + SQL (Structured Query Language)
  + ER (Entity Relationship)
* Technical words with definition
  + Database Schema: The structure or blueprint of a database.
  + Logical Schema: Abstract representation of data organization in tables and their relationships.
  + Physical Schema: How data is actually stored on disk (database-specific implementation).
  + Entity: A real-world object or concept represented in a database (e.g., customer, order).
  + Attribute: A characteristic of an entity (e.g., customer name, order date).
  + Primary Key: A field or combination of fields that uniquely identifies each record in a table.
  + Foreign Key: A field in a table referencing a primary key in another table, establishing a relationship.
* Software Packages
  + MySQL (implied): A popular relational database management system.
* Commands/modules with definition
  + CREATE TABLE table\_name (...): Creates a new table (physical schema).
* Ideas/techniques with definition
  + ER Modeling (Entity Relationship Modeling): A way to graphically design database schemas, visualizing entities and their relationships.
* keywords = ['SELECT', 'FROM', 'WHERE', 'CREATE TABLE', 'PRIMARY KEY', 'FOREIGN KEY']

Mistakes or Misleading Information:

* **Oversimplifies Physical Implementation:** Database systems employ complex indexing, storage, and optimization techniques beyond simple disk storage.

Written by Gemini

[Building a schema | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/ElOsG/building-a-schema) 3-26-2024 1009

Short Summary: This reading explains how to design a database schema for a restaurant booking system. It covers the concepts of logical and physical schemas, tables, columns, data types, and establishing relationships between tables.

**Acronyms**

* **SQL:** Structured Query Language
* **DDL:** Data Definition Language
* **ERD / ER-D:** Entity-Relationship Diagram

**Terms**

* **Database:** A structured collection of data stored electronically.
* **Database Schema:** A blueprint or outline of a database, defining its structure.
* **Schema Objects:** Elements within a database schema, such as tables, columns, relationships, data types, etc.
* **Table:** A structure within a database that holds data in rows and columns.
* **Column:** A vertical field in a table that holds a specific type of data.
* **Data Type:** Determines the kind of values a column can store (e.g., INT for integers, VARCHAR for text).
* **Primary Key** A unique identifier for each row (record) in a table.
* **Foreign Key** A column in a table that references the primary key of another table, creating a relationship between them.

**Technical Words**

* **Restaurant Booking System:** A software system for managing restaurant reservations.
* **Entity:** A real-world object or concept represented in a database (e.g., a customer, a table).
* **Attribute:** A property or characteristic of an entity (e.g., a customer's name).
* **Relationship:** A connection or association between different entities in a database.
* **Reservation:** The act of booking a table at a restaurant.
* **Menu:** A list of food and drinks available at a restaurant.

**Rarely Used Words**

* **NIC Number:** National Identity Card number (this might be regionally specific)
* **Pax:** Abbreviation for "passengers," often used in the hospitality industry to refer to the number of guests.

**Software**

* **MySQL:** A popular open-source relational database management system.

**Commands**

* **CREATE DATABASE:** SQL command to create a new database. **CREATE DATABASE pet\_clinic;**
* **CREATE TABLE**: SQL command to create a new table within a database.
  + - **CREATE TABLE dogs (**
    - **dog\_id INT PRIMARY KEY,**
    - **name VARCHAR(50),**
    - **breed VARCHAR(50),**
    - **birthdate DATE**
    - **);**
* -- First switch to the database you just created **USE pet\_clinic;**

**Ideas**

* **Database Design:** The process of planning and defining the structure of a database.
* **Logical Schema:** A conceptual model of a database, focusing on entities and their relationships.
* **Physical Schema:** The implementation of a logical schema in a specific database system, including details like data types.

**Important Notes**

* Primary keys ensure uniqueness and are essential for creating relationships between tables.
* Foreign keys establish links between related tables.
* Careful schema design is crucial for a well-organized and efficient database.

**Keywords:** database, schema, table, column, data type, primary key, foreign key, relationship, SQL, CREATE DATABASE, CREATE TABLE

* Notes taken by Gemini
* I used **MySQL Workbench**. It is in the MySQL workbench directory is you did the full install. Select your DB.

A screenshot of a computer

Description automatically generated

* Look in the Database section and select reverse engineer or ctrl-R.

A screenshot of a computer

Description automatically generated

[Knowledge check: Database schema | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/EP3Aj/knowledge-check-database-schema) 3-26-1200

* This test was not hard, but I still missed one. Ill try again soon.

[Table relationships | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/faeqV/table-relationships) 3-26-2024 1225

Short Summary: This transcript explains the relational database model, focusing on how different relationship types (one-to-many, one-to-one, many-to-many) connect tables within a database. It also introduces Entity-Relationship Diagrams (ERDs) as a way to visualize these relationships.

**Acronyms**

* **ERD:** Entity Relationship Diagram
* **PK:** Primary Key
* **FK:** Foreign Key

**Terms**

* **Database:** A structured collection of data stored electronically.
* **Relational Model:** A way of organizing data in a database using tables with relationships between them.
* **Table:** A structure within a database that holds data in rows and columns.
* **Column:** A vertical field in a table that holds a specific type of data.
* **Row (Record):** A single horizontal entry in a table.

**Technical Words**

* **Entity:** A real-world object or concept represented in a database (e.g., student, course, department).
* **Relationship:** A connection between entities in a database.
* **Primary Key:** A unique identifier for each row (record) in a table.
* **Foreign Key:** A column in a table that references the primary key of another table, creating a relationship.

**Ideas**

* **One-to-Many Relationship:** One record in a table can be linked to multiple records in another table (e.g., one student can take many courses).
* **One-to-One Relationship:** One record in a table is linked to only one corresponding record in another table (e.g., one department head leads one department).
* **Many-to-Many Relationship:** Multiple records in one table can be linked to multiple records in another table (e.g., many students can work on many research projects).
* **Entity-Relationship Diagrams (ERDs):** Visual representations of a database schema, showing entities, attributes, and relationships.

**Keywords:** database, relational model, table, column, row, record, entity, relationship, primary key, foreign key, one-to-many, one-to-one, many-to-many, ERD

Notes taken by Gemini

[Relational model | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/4jtgY/relational-model) 3-26-2024 1215

Short Summary: This reading explains the fundamental concepts of the relational database model. It discusses tables (relations), columns, rows, keys, constraints, and the different types of relationships that can exist between tables (one-to-one, one-to-many, many-to-many).

**Terms**

* **Relational Database Model:** A way of organizing data into tables with defined relationships, forming the basis for most modern database systems.
* **Table (Relation):** A structure within a database that stores data in rows and columns.
* **Row (Tuple, Record):** A single horizontal entry in a table representing a group of related data.
* **Column (Field, Attribute):** A vertical field within a table that holds a specific type of data.
* **Data Type:** The classification of the kind of values a column can store (e.g., text, numbers, dates).
* **Domain:** The set of acceptable values that a column can contain, determined by its data type.
* **Key:** One or more columns that uniquely identify each row in a table.
* **Primary Key:** The main unique identifier for each row in a table.
* **Foreign Key:** A column in a table that references the primary key of another table, establishing a relationship.
* **Degree:** The number of columns in a table.
* **Cardinality:** The number of rows in a table.

**Technical Words**

* **Database:** A structured collection of data stored electronically.
* **SQL:** (Structured Query Language) The standard language used to interact with relational databases.

**Ideas**

* **Relational Integrity Constraints:** Rules that ensure the validity and consistency of data within a relational database:
  + **Key constraints:** Enforce uniqueness and prevent null values in primary keys.
  + **Domain constraints:** Ensure data values match their defined data types.
  + **Referential integrity constraints:** Guarantee that foreign keys match existing values in the tables they reference.
* **Types of Relationships**
  + **One-to-One (1:1):** Each record in one table corresponds to exactly one record in another table.
  + **One-to-Many (1:N):** One record in a table can be linked to multiple records in another table.
  + **Many-to-Many (N:N):** Multiple records in one table can be linked to multiple records in another table (often modeled using a junction table).

**Keywords:** relational database model, table, column, row, record, data type, domain, key, primary key, foreign key, constraint, relationship, one-to-one, one-to-many, many-to-many

Notes taken by Gemini

[Primary key | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/Pxl2m/primary-key) 3-26-2024 1230

Short Summary: This transcript explains primary keys in databases. It discusses how primary keys ensure the uniqueness of records, prevent duplicates, and facilitate the retrieval of specific data, including the concept of composite primary keys.

**Terms**

* **Database Table:** A structure within a database that stores data in rows and columns.
* **Record (Row):** A single horizontal entry in a table representing a group of related data.
* **Column (Attribute):** A vertical field in a table that holds a specific type of data.
* **Primary Key:** A column (or combination of columns) that uniquely identifies each record in a table.
* **Null Value:** Represents missing or unknown data.
* **Candidate Key:** An attribute (column) that could potentially serve as a primary key.
* **Alternate Key (Secondary Key):** A candidate key that was not chosen as the primary key.
* **Composite Primary Key:** A primary key created by combining two or more columns to ensure uniqueness.

**Technical Words**

* **Database:** A structured collection of data stored electronically.

**Ideas**

* **Uniqueness:** Primary keys guarantee that each record in a table has a unique identifier.
* **Data Retrieval:** Primary keys make it easy to locate and access specific records within a table.
* **Composite Keys:** In situations where no single column has unique values, combining multiple columns can create a unique identifier.

**Keywords:** database, table, record, column, primary key, candidate key, composite key, unique, null value

Notes taken by Gemini

[Foreign key | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/SrBoS/foreign-key) 3-26-2024 1310

Short Summary: This reading explains how to select primary keys in database tables and how to connect tables using foreign keys. It includes practical SQL examples for creating tables and establishing relationships.

**Terms**

* **Database:** A structured collection of data stored electronically.
* **Primary Key:** A unique identifier for each record in a table.
* **Candidate Key:** An attribute (column) that could potentially serve as a primary key.
* **Alternate Key:** A candidate key not chosen as the primary key.
* **Foreign Key:** A column in a table that references the primary key of another table, creating a relationship between them.
* **Composite Key:** A primary key composed of multiple columns.
* **Parent Table:** The table containing the primary key that a foreign key references.
* **Child Table:** The table containing the foreign key that references a parent table.

**Technical Words**

* **Database Management System (DBMS):** Software for creating and managing databases (e.g., Oracle, MySQL).
* **SQL (Structured Query Language):** The standard language for interacting with relational databases.
* **Entity Relationship Diagram (ERD):** A visual representation of a database schema.

**Rarely Used Words**

* **Cross-reference:** Connecting related information within a database.

**Software**

* **MySQL** A popular open-source relational database management system.

**Commands**

* **CREATE DATABASE:** SQL command to create a new database.
* **USE:** SQL command to select a specific database.
* **CREATE TABLE:** SQL command to create a new table.
* **SHOW TABLES:** SQL command to list tables in a database.
* **SHOW COLUMNS FROM:** SQL command to display the structure of a table.
* **ALTER TABLE ... ADD FOREIGN KEY ... REFERENCES:** SQL command to define a foreign key and establish a relationship between tables.

**Ideas**

* **Primary Key Selection:** Primary keys should be unique, unchanging, and ideally simple.
* **Table Relationships:** Foreign keys create connections between tables, allowing for data association.
* **One-to-Many Relationship:** One record in a parent table can be associated with multiple records in a child table.

**Keywords:** database, table, primary key, candidate key, foreign key, SQL, CREATE DATABASE, CREATE TABLE, SHOW TABLES, SHOW COLUMNS, ALTER TABLE

Notes taken by Gemini

A screenshot of a computer

Description automatically generated

In the provided entity-relationship diagram (ERD), we observe two tables that articulate a one-to-many association, integral to the domain of vehicle ownership management. The owner table serves as the parent entity, encapsulating essential attributes of vehicle proprietors, notably a uniquely identifiable ownerID, the ownerName, and the ownerAddress. This primary key, ownerID, is of critical importance as it establishes referential integrity within the database.

The vehicle table is delineated as a child entity containing details specific to individual vehicles. It includes a primary key vehicleID, the corresponding ownerID as a foreign key, the plateNumber for registration identification, and a phoneNumber for contact purposes. The foreign key ownerID in the vehicle table creates a direct relational link to the owner table, enforcing the relationship whereby a single owner can be associated with multiple vehicles, yet each vehicle is definitively linked to a solitary owner.

The ERD's depiction of the relationship employs standard notation, with a singular line emanating from the owner table, devoid of crow's foot, merging with a line that fans out into several branches as it approaches the vehicle table—symbolically representing the "many" endpoint of this one-to-many relationship.

This relationship is fundamental to the relational database schema as it allows for the organization of complex and related data in a manner that is both efficient for storage and optimal for query performance. Proper indexing strategies, while not detailed in this diagram, would be essential to support high-performance searches and data retrieval operations across these tables.

Description by ChatGPT / Diagram by me via MySQL Workbench

[Knowledge check: Defining keys | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/uPZoU/knowledge-check-defining-keys) 3-36-2024 1335

* Basic quiz, missed one but only because I went to fast and did not read it.

[Finding entitites | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/TtbV4/finding-entitites) 3-26-2024 1350

Short Summary: This transcript explains entities and attributes within relational databases. Entities represent objects or concepts stored in the database, while attributes describe their characteristics.

**Terms**

* **Entity:** An object, person, place, or concept about which data is stored in a database.
* **Attribute:** A specific type of data associated with an entity in a relational database. Each attribute is represented by a column in a table.
* **Relational Database:** A database model based on organizing data into tables and the relationships between them.
* **Table:** A structure within a database that stores data in rows and columns.
* **Row (Record):** A single horizontal entry in a table representing an instance of an entity.
* **Column (Field):** A vertical element in a table that holds a specific type of data.
* There are subtle differences between the terms "column," "field," and "attribute" in the context of databases, although they are often used interchangeably:
* **Attribute:** The most abstract of the three terms. An attribute is a conceptual characteristic or property of an entity (a thing you want to store data about). It defines the type of data that will be stored.
* Example: In designing a "Customer" database, you might define attributes like "Name," "Email," and "Phone Number."
* **Column:** The physical implementation of an attribute within a database table. A column has a name, a data type, and holds the actual data values in each row.
* Example: When you create the "Customer" table, you'll have columns named "Name," "Email," and "Phone Number," with data types like text and number.
* **Field:** A somewhat flexible term that can be used in different ways:
* **Synonym for Column:** In some contexts, "field" is interchangeable with "column," referring to a specific element within a table's row.
* **Form Input:** "Field" often refers to a data entry box on a user interface form, where a user might enter their name, email, etc. This data would then be mapped to the corresponding columns in the database.
* **Summary**
* Technically, attributes define columns, and columns can be referred to as fields.
* In everyday database discussions, the terms are often used interchangeably, so the context will determine the precise meaning.

**Technical Words**

* **Database Management System (DBMS):** Software for creating and managing databases (e.g., MySQL).

**Types of Attributes**

* **Simple attribute:** An attribute that cannot be further subdivided.
* **Composite attribute:** An attribute that can be broken into smaller components.
* **Single-valued attribute:** An attribute that can hold only one value.
* **Multi-valued attribute:** An attribute that can hold multiple values.
* **Derived attribute:** An attribute whose value is calculated from other attributes.
* **Key attribute** A unique identifier for each record of an entity.

**Keywords:** entity, attribute, relational database, table, row, column, simple attribute, composite attribute, single-valued attribute, multi-valued attribute, derived attribute, key attribute

Notes taken by Gemini

[Entity relationship diagrams (ERD) | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/supplement/9AWeu/entity-relationship-diagrams-erd) 3-26-2024 1400

* Summary: This text explains the use of Entity Relationship Diagrams (ERDs) in designing relational databases, emphasizing the identification and documentation of entities, attributes, and their relationships.
* Acronyms:
* ERD: Entity Relationship Diagram
* SDLC: Software Development Life Cycle
* Terms:
* Entity: A real-world object or concept represented within a database.
* Attribute: A property or characteristic of an entity.
* Relationship: The association between entities within a database.
* Cardinality: The numerical relationship between two entities in a relationship.
* Foreign Key: A field in one table that uniquely identifies a row of another table.
* Technical words:
* Relational Database Model: A type of database structure where data is stored in tables related by common fields.
* Data Type: A classification specifying the type of data that a variable or database field can hold.
* Software packages:
* Oracle: A multi-model database management system produced and marketed by Oracle Corporation.
* MySQL: An open-source relational database management system.
* Database Design: The process of producing a detailed data model of a database.
* Documentation: The written details that explain how software or a system works or how it was developed.
* Implementation: The process of putting a decision or plan into effect; in databases, it's about creating the actual database from the ERD.
* Techniques:
* Entity Relationship Modelling: A methodology for creating ERDs to represent the data requirements of a business.
* Keywords = ["ERD", "Relational Database", "Entity", "Attribute", "Relationship", "Cardinality", "Foreign Key", "Oracle", "MySQL", "Database Design", "Documentation", "Implementation"]

Written by ChatGPT.

[Database relations and keys | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/ruGop/database-relations-and-keys) 3-26-2024 1420

Short Summary: This transcript explains database normalization, a process that optimizes database design by reducing redundancy and making data management easier. It discusses problems like insert, update, and delete anomalies that can occur in poorly designed databases.

**Terms**

* **Database Normalization:** The process of structuring a database to reduce data redundancy and improve data integrity.
* **Database Table:** A structure within a database that stores data in rows and columns.
* **Primary Key:** A unique identifier for each record (row) in a table.
* **Anomaly:** An unexpected behavior in a database that can lead to data inconsistency.

**Technical Words**

* **Database System:** A system used to manage and store data in a structured form.
* **SQL (Structured Query Language):** The standard language for interacting with relational databases.

**Types of Anomalies**

* **Insert Anomaly:** Occurs when you cannot insert a new record due to missing data in other required fields.
* **Update Anomaly:** Occurs when changing data in one part of a table necessitates updates in multiple other places to maintain consistency.
* **Deletion Anomaly:** Occurs when deleting a record also removes unrelated data that should be retained.
* keywords = ["database", "normalization", "table", "primary key", "anomaly", "insert anomaly", "update anomaly", "deletionanomaly", "SQL"]

[What is database normalization? | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/fuQas/what-is-database-normalization) 3-26-2024 1500

Notes taken by Gemini

Short Summary: This reading explains database normalization, a process of restructuring tables to minimize data redundancy, improve data integrity, and simplify queries. It focuses on the first three normal forms (1NF, 2NF, 3NF) and demonstrates their application through an example.

**Terms**

* **Database Normalization:** The process of structuring a database to reduce data redundancy and improve data integrity.
* **First Normal Form (1NF):** Requires each table cell to hold a single value and eliminates repeating groups of data.
* **Second Normal Form (2NF):** Builds upon 1NF and requires all non-key attributes to depend on the entire primary key, not just part of it.
* **Third Normal Form (3NF):** Extends 2NF and prohibits transitive dependencies, where a non-key attribute depends on another non-key attribute.
* **Primary Key:** A unique identifier for each record in a table.
* **Foreign Key:** A column in a table that references the primary key of another table.
* **Partial Dependency:** A relationship in a table where a non-key attribute depends on only part of a composite primary key.
* **Transitive Dependency:** A relationship in a table where a non-key attribute depends on another non-key attribute.

**Technical Words**

* **Database:** A structured collection of data stored electronically.
* **Table:** A structure within a database that stores data in rows and columns.
* **Composite Key:** A primary key composed of multiple columns.
* **Atomicity:** The principle that each cell in a table should hold a single value.
* **SQL (Structured Query Language):** The standard language for interacting with relational databases.

**Keywords** keywords = ["database", "normalization", "table", "primary key", "foreign key", "1NF", "2NF", "3NF", "partial dependency", "transitive dependency", "atomicity", "SQL"]

Notes taken by Gemini

[First normal form 1NF | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/YEH1G/first-normal-form-1nf) 3-26-1510

Short summary: This video explains database normalization, focusing on achieving first normal form (1NF). 1NF requires single values in each field (atomicity) and eliminates repeating data groups to improve efficiency and data integrity.

**Acronyms**

* 1NF: First Normal Form
* 2NF: Second Normal Form
* 3NF: Third Normal Form

**Terms**

* Database: A structured collection of data.
* Table: A way to organize data using rows and columns.
* Column: A vertical set of data values within a table, representing an attribute.
* Field: A single data point at the intersection of a row and column in a table.
* Row: A horizontal set of data values within a table, representing a single record.
* Primary Key: A unique identifier for each row in a table.
* Foreign Key: A column in one table that references the primary key in another table, establishing a relationship between them.

**Technical Words**

* Normalization: The process of organizing data in a database to reduce redundancy and improve data integrity.
* Data Atomicity: The principle that each field in a database table should contain only one indivisible value.
* Repeating Groups: Sets of data that repeat within a single table row, violating atomicity.
* Data Redundancy: Storing the same data in multiple places, potentially leading to inconsistencies.
* Data Integrity: The accuracy and consistency of data throughout its lifecycle.

**Rarely Used Words**

* Anomaly: Problem or inconsistency in a database that can arise due to poor design.

**Important Ideas**

* Database normalization is essential to create well-structured, efficient, and maintainable databases.
* First Normal Form (1NF) is the starting point of normalization, ensuring data atomicity and eliminating repeating groups.
* Achieving 1NF involves breaking down tables into smaller, related tables and using foreign keys to link them.

**Techniques**

* Identifying repeating groups of data.
* Creating separate tables for each entity (e.g., courses, tutors).
* Using primary and foreign keys to link related tables.

**Keywords** ["database", "table", "column", "field", "row", "primary key", "foreign key", "normalization", "data atomicity", "repeating groups", "data redundancy", "data integrity"]

This analysis was provided by Gemini, a large language model from Google AI.

[Second normal form 2NF | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/dtFfc/second-normal-form-2nf) 3-26-2024 1529

Short summary: This video explains second normal form (2NF) in database normalization. 2NF builds on first normal form (1NF) to eliminate partial dependencies, ensuring non-key attributes depend on the entire primary key, improving database structure.

**Acronyms**

* 1NF: First Normal Form
* 2NF: Second Normal Form

**Terms**

* Database: A structured collection of data.
* Table: A way to organize data using rows and columns.
* Column: A vertical set of data values within a table, representing an attribute.
* Field: A single data point at the intersection of a row and column in a table.
* Row: A horizontal set of data values within a table, representing a single record.
* Primary Key: A unique identifier for each row in a table.

**Technical Words**

* Database normalization: The process of organizing data in a database to reduce redundancy and improve data integrity.
* Relational database: A type of database that organizes data into related tables.
* Composite primary key: A primary key made up of two or more columns.
* Functional dependency: A relationship where the value of one attribute in a table determines the value of another attribute.
* Partial dependency: In a table with a composite primary key, a non-key attribute is partially dependent if it depends on only one part of the primary key.

**Rarely Used Words**

* Anomaly: Problem or inconsistency in a database that can arise due to poor design.
* Retrieve: To fetch or get data from a database.

**Ideas**

* Database normalization: A step-by-step process to improve database design.
* First normal form (1NF): A table is in 1NF if all attributes have atomic values and there are no repeating groups.
* Second normal form (2NF): A table is in 2NF if it is in 1NF and all non-key attributes are fully dependent on the entire primary key.

**Techniques**

* Identifying functional and partial dependencies.
* Splitting tables into smaller tables based on entities.
* Using primary keys to establish relationships between tables.

**Keywords** ["database", "table", "column", "field", "row", "primary key", "normalization", "relational database", "composite primary key", "functional dependency", "partial dependency", "1NF", "2NF"]

This analysis was provided by Gemini, a large language model from Google AI.

[Knowledge Check: Database normalization | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/W5LLk/knowledge-check-database-normalization) 3-26-2024 1537

* Missed a few and I do not feel I grasp this fully so redo test.

[Third normal form 3NF | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/ctzx9/third-normal-form-3nf) 3-26-2024 1543

Short summary: This video explains third normal form (3NF) in database normalization, focusing on eliminating transitive dependencies. Transitive dependencies occur when non-key attributes depend on each other, violating 3NF rules.

**Acronyms**

* 3NF: Third Normal Form

**Terms**

* Database: A structured collection of data.
* Table: A way to organize data using rows and columns.
* Column: A vertical set of data values within a table, representing an attribute.
* Field: A single data point at the intersection of a row and column in a table.
* Row: A horizontal set of data values within a table, representing a single record.
* Primary Key: A unique identifier for each row in a table.
* Foreign key: A column in one table that references the primary key in another table, establishing a relationship between them.

**Technical Words**

* Database normalization: The process of organizing data in a database to reduce redundancy and improve data integrity.
* Relational database: A type of database that organizes data into related tables.
* Transitive dependency: When a non-key attribute depends on another non-key attribute, violating third normal form.
* Key attribute: An attribute in a database that helps to uniquely identify a row of data in a table.

**Rarely Used Words**

* Context: The setting or circumstances in which something occurs.

**Ideas**

* Database normalization: A step-by-step process to improve database design.
* First normal form (1NF): A table is in 1NF if all attributes have atomic values and there are no repeating groups.
* Second normal form (2NF): A table is in 2NF if it is in 1NF and all non-key attributes are fully dependent on the entire primary key.
* Third normal form (3NF): A table is in 3NF if it is in 2NF and there are no transitive dependencies.

**Techniques**

* Identifying transitive dependencies.
* Splitting tables to remove transitive dependencies and create relationships between tables.
* Using foreign keys to link related tables.

**Important People**

* Cormac O'Dwyer: Fictional author used as an example.
* Michel Leiris: Fictional author used as an example. Note: Michel Leiris was a real French writer, but this video uses him as a fictional example.

**Keywords** ["database", "table", "column", "field", "row", "primary key", "foreign key", "normalization", "relational database", "transitive dependency", "3NF"]

This analysis was provided by Gemini, a large language model from Google AI.

[Database schema examples | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/ungradedLab/eXuXV/database-schema-examples) 3-26-2024 1600

* Basic

[Self-review: Database schema examples | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/quiz/uSv0P/self-review-database-schema-examples/attempt?redirectToCover=true) 3-26-2024 1610

Short quiz. Missed one. Try this again.

[Module summary: Database design | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/1jZOS/module-summary-database-design) 3-26-2024 1613

* Just a summary

[Module summary: Database design | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/1jZOS/module-summary-database-design)

* Summary

[Course Recap: Introduction to databases for back-end development | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/lecture/CgBAo/course-recap-introduction-to-databases-for-back-end-development)

* Review

[Final graded quiz: Intro to databases | Coursera](https://www.coursera.org/learn/intro-to-databases-back-end-development/exam/zxxQm/final-graded-quiz-intro-to-databases/view-attempt) 3-26-2024 2025

* 95% Way to many weird syntax errors that appear correct.