DDL – Data Definition Language

Data Definition Language - DDL

- Creating, modifying, deleting
  - **Database**
  - **Table**
  - Index
  - View

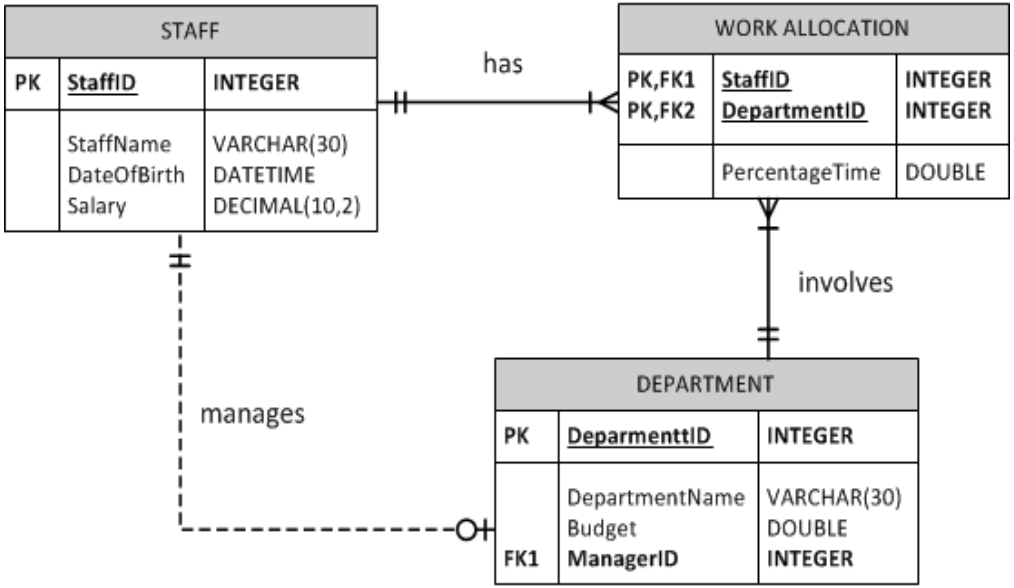
Steps for creating a new database

1. Create Database
2. Create Tables
3. Insert data into tables (DML)

Commands include:

- **CREATE**
- ADD
- ALTER
- DROP

ERD (Physical model) and Relation Schema



Business rule:

- An employee may work in several departments, with the percentage of time spent in each department being recorded in the **WORK ALLOCATION** table

STAFF(StaffID, StaffName, DateOfBirth, Salary)

DEPARTMENT(DepartmentID, DepartmentName, Budget, *ManagerID*)

WORK ALLOCATION(*StaffID*, *DepartmentID*, PercentageTime)

Relational Database Schema

Tables	Attributes	Data types	Description
STAFF	StaffID	INT	PRIMARY KEY
	Staffname	VARCHAR(30)	
	DateOfBirth	Date	
	Salary	DECIMAL(10,2)	
DEPARTMENT	DepartmentID	INT	PRIMARY KEY
	DepartmentName	VARCHAR(30)	
	Budget	DOUBLE	
	ManagerID	INT	FK –REFERENCES STAFF(StaffID)
WORK ALLOCATION	StaffID	INT	PK – FK REFERENCES STAFF(StaffID)
	DepartmentID	INT	PK – FK REFERENCES DEPARTMENT(DepartmentID)
	PercentageTime	DOUBLE	

Step 1 – Create a new database

- Syntax:

  CREATE DATABASE [IF NOT EXISTS] ***db_name;***

- Example:

  CREATE DATABASE
  DB_Week7;

  or

  CREATE DATABASE **IF NOT EXISTS**
  DB_Week7;

- USE DB_Week7;
- SHOW TABLES;

- The IF NOT EXISTS option is useful if you are using a script to create a database. This determines if it already exists, if it does not the database is created.
- If you are doing this on the command line you should then use the USE command to indicate that you will be working with the new database.
- What does SHOW TABLES result?

Step 2 – Create the Tables

- Syntax:

```
CREATE TABLE [IF NOT EXISTS ] tablename
    (column1 datatype [PRIMARY KEY],
    column2 datatype [NOT NULL],
    column3 datatype [DEFAULT expr],
    column4 datatype [UNIQUE],
    column5 datatype [CHECK expr],
     …,
    columnX datatype [REFERENCES parent_table(PK attribute)]);
```

- IF NOT EXISTS **– prevents recreating an existing table**
- You must specify the *tablename*, *column* name, column *datatype* and size
- Each column specification starts with a column name, followed by a datatype
- You can specify a number of constraints (restrictions) for a column or the entire table
- **Order is important here** – particularly for tables with relationships between them. In one to many relationship you need to create the tables with the 'one' side of relation before the 'many' side

Step 2 – Create the Tables

- The following creates the STAFF table, with primary keys and data types

```
CREATE TABLE IF NOT EXISTS STAFF (
    StaffID          INT PRIMARY KEY AUTO_INCREMENT,
    StaffName        VARCHAR(30),
    DateOfBirth      DATE,
    Salary           DECIMAL(10,2)
) ENGINE = InnoDB;
```

- **InnoDB** is a transaction-safe storage engine that supports foreign key referential-integrity constraints to maintain data integrity.

Step 2 – Create the Tables

- The following creates the DEPARTMENT table, with primary keys, data types and relation information (FK)

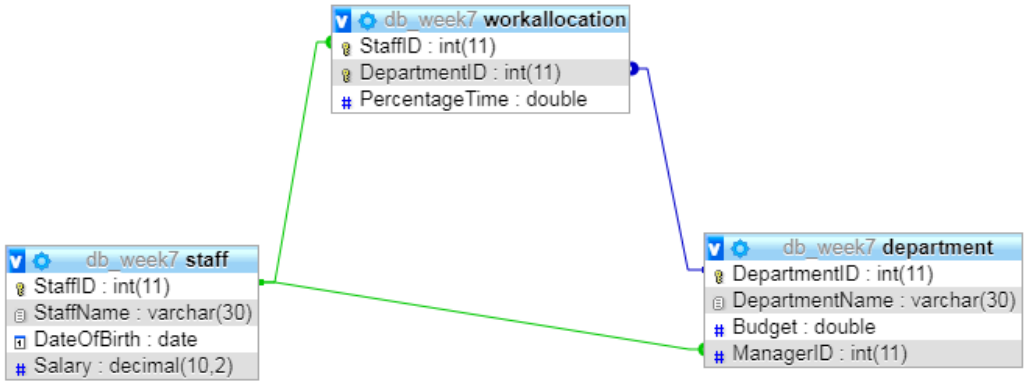
```
CREATE TABLE IF NOT EXISTS DEPARTMENT(
    DepartmentID        INT PRIMARY KEY AUTO_INCREMENT,
    DepartmentName      VARCHAR(30),
    Budget                DOUBLE,
    ManagerID           INT NOT NULL,
    FOREIGN KEY (ManagerID) REFERENCES Staff(StaffID)
) ENGINE=InnoDB;
```

- **'NOT NULL' is used when data must be provided for the field**
  - In ERD it is because of MANDATORY sign (Modality)
  - E.g., at the STAFF side we have mandatory sign which indicates Department must have a manager!
- **FOREIGN KEY** = to make relationship between tables, sets up FK for the relationship

Step 2 – Create the Tables

- The following creates the WORK ALLOCATION table, with primary keys and data types

```
CREATE TABLE  IF NOT EXISTS
WORKALLOCATION(
    StaffID             INT,
    DepartmentID        INT,
    PercentageTime      DOUBLE,
    PRIMARY KEY (StaffID, DepartmentID),
    FOREIGN KEY (StaffID) REFERENCES STAFF(StaffID),
    FOREIGN KEY (DepartmentID) REFERENCES DEPARTMENT(DepartmentID)
) ENGINE = InnoDB;
```



Thank you