# Name of Your Company

**Sitters** “bringing families closer to nannies.”

# Project Title

Database for a company that provides babysitting services to families.

# Team

* Shedrach Elurihu Ezenwali (MySQL Expert)
* Al Ghandour Muhammad (MS SQL Server Expert)

The screenshot of both DBMS is shown below.

A computer screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

# Weekly Meeting Hours

We will meet and work on the project every Thursday after the class for one hour (15:30 – 16:30) at the Anvil Tower. Further communication about the project will be done through phone or messages.

# Project Description

The database stores data about Users (nannies, families), and details about the contract. Information about the nanny, like name, date of birth, address, phone, language, Reviews, highest education degree, gender, occupation, availability, SSN, skills, and nationality would be saved. For the family, name, address, marital status, phones, emails, languages spoken, **Child information** (name, date of birth, age, gender, disability, languages spoken, preferences) would be collected and saved. The contract between the family and the nanny will store start and end datetime, payment, and notes on instruction. We might also save reviews and comment after contract has ended.

# Assumptions about Cardinality and Participations

You can write all the assumptions about Cardinality and Participations (total/partial) here.

* A family can book many Nannies.
* A Nanny can have many bookings from different families.
* Participation on the contract is total for both users (Families and nannies).
* Each family must have at least a child. And a child can only belong to one family.
* A family can have more than one child that needs a nanny and can assign them to different nannies.

# EER Modeling Diagram

In the following drawing canvas, EER Modeling shapes have been provided. You can copy and replicate them (Ctrl+C to copy and Ctrl+V to paste. You can also select a shape, then press Ctrl button and drag and drop to copy a shape) and edit them to build your diagram.

A group of white ovals with black text

Description automatically generated

# ER-Model Mapping to Database Relational Schema

User(userID, email, address, fullname, birthdate, mobile\_number )

Family\_rep(**userID**, occupation, marital\_status)

Nanny(**userID**, ssn, highest\_edu, gender, availability)

Nanny\_skill(**userID**,skill)

Contract(contractID, start\_date, end\_date, pay\_per\_hour)

Contract\_schedule(**contractID**, start\_date\_time, end\_date\_time, description)

Child(child\_key, **userID,** birth\_date, child\_name, gender)

Child\_Preferences(**child\_key**, **userID,** preference)

Child\_Disability(**child\_key**, **userID,** disability)

Family\_rep\_sign\_contract(**child\_key**, **userID,contractID**)

Nanny\_sign\_contract(**userID, contractID**)

# Normalization

All relations must be normalized up to BCNF. You must explain why you believe every relation in your database in normalized.

User(userID, email, address, fullname, birthdate, mobile\_number )

**First Normal Form (1NF):**

Already in 1NF as it has a primary key (userID) and atomic values in each column.

### Second Normal Form (2NF):

Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Family\_rep(userID, occupation, marital\_status)

### First Normal Form (1NF):

Already in 1NF. It has a primary key (userID) and atomic values.

### Second Normal Form (2NF):

Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Nanny(userID, ssn, highest\_edu, gender, availability)

### First Normal Form (1NF):

### Already in 1NF. It has a primary key (userID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

### Already in 3NF. No transitive dependencies.

Nanny\_skill(userID,skill)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (userID, skill).

### Second Normal Form (2NF):

### Already in 2NF. The composite primary key (userID, skill) covers all non-prime attributes.

### Third Normal Form (3NF):

### Already in 3NF. No transitive dependencies.

Contract(contractID, start\_date, end\_date, pay\_per\_hour)

### First Normal Form (1NF):

### Already in 1NF. It has a primary key (contractID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Contract\_schedule(contractID, start\_date\_time, end\_date\_time, description)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (contractID, start\_date\_time, end\_date\_time) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Child(child\_key, userID, birth\_date, child\_name, gender)

### First Normal Form (1NF):

### Already in 1NF. It has a primary key (child\_key) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Child\_Preferences(child\_key, userID, preference)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (child\_key, userID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. The composite primary key (child\_key, userID) covers all non-prime attributes.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Child\_Disability(child\_key, userID, disability)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (child\_key, userID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. The composite primary key (child\_key, userID) covers all non-prime attributes.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Family\_rep\_sign\_contract(child\_key, userID,contractID)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (child\_key, userID, contractID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

Nanny\_sign\_contract(userID, contractID)

### First Normal Form (1NF):

### Already in 1NF. It has a composite primary key (userID, contractID) and atomic values.

### Second Normal Form (2NF):

### Already in 2NF. No partial dependencies.

### Third Normal Form (3NF):

Already in 3NF. No transitive dependencies.

**Boyce-Codd Normal Form (BCNF):**

* **All relations are already in BCNF:**
  + Every determinant is a candidate key in each table. There are no non-trivial dependencies on superkeys.

# Determining Data Types (Domain) and Constraints

You explain why you choose a certain data type for a field and why you apply certain constraints

# Creating Database and Tables - SQL DDL

You do not need to copy SQL commands here. Save your SQL commands in a script file and just mention the name of the file here. Make sure the script file is stored besides this document within the same folder.

# Inserting Values in Tables

You do not need to copy SQL commands here. Save your SQL commands in a script file and just mention the name of the file here. Make sure the script file is stored beside this document within the same folder.

# SQL Queries

You do not need to copy SQL commands here. Save your SQL commands in a script file and just mention the name of the file here. Make sure the script file is stored beside this document within the same folder.

# Views

You do not need to copy SQL commands here. Save your SQL commands in a script file and just mention the name of the file here. Make sure the script file is stored beside this document within the same folder.