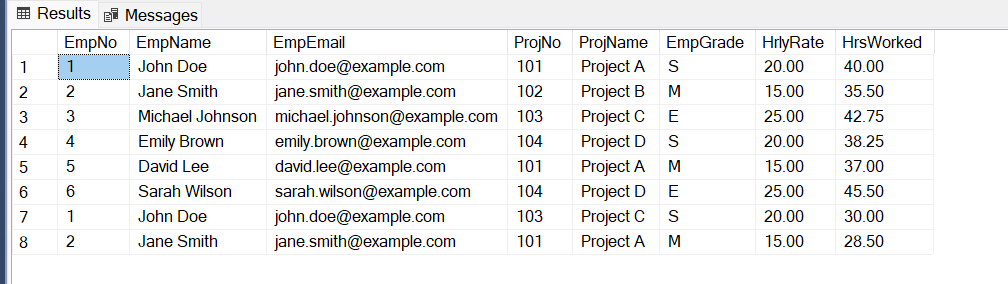
W1D5: Assignment 3 – Normalization

## Employee , Project Relation .

1. **Table Creation and Sample Data (SQL Server):**



**b. Functional Dependencies (FDs):**

* EmpNo -> EmpName, EmpEmail, EmpGrade, HrlyRate (unique attribute determines all other employee attributes)
* ProjNo -> ProjName (unique attribute determines project name)
* EmpGrade -> HrlyRate (employee grade determines hourly rate)
* (EmpNo, ProjNo) -> HrsWorked (combination of employee and project uniquely determines hours worked)

**c. Candidate Keys (CKs):**

* {EmpNo} (uniquely identifies an employee)
* {ProjNo} (uniquely identifies a project)
* {(EmpNo, ProjNo)} (uniquely identifies an employee's work on a specific project)

**d. Primary Key (PK):**

* **{EmpNo, ProjNo}:** Uniquely identifies each employee's work on specific projects.

**e. Partial Dependencies:**

* EmpNo -> EmpName, EmpEmail, EmpGrade, HrlyRate .
* ProjNo -> ProjName .

**f. Normalization to 2NF:**

The presence of a determining attribute (EmpNo, ProjNo) for each non-key attribute (EmpName, EmpEmail, EmpGrade, HrlyRate) satisfies 2NF.

**g. Tables in 2NF (No changes required):**

* The table remains the same as in step (a) since it already meets the 2NF criteria.

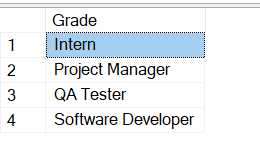
**h. Normalization to 3NF:**

The transitive dependency violates 3NF because the non-key attribute HrlyRate is dependent on the non-key attribute EmpGrade, which is not part of the primary key.

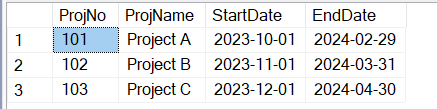
empNo -> empGrade -> hrlRate

To achieve 3NF, we need to decompose the table into the following separate tables:

GradeRate  
empGrade, hrlRate

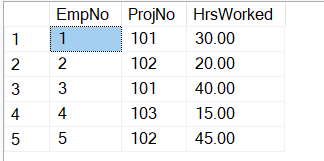


ProjectTable  
proNo, ProName



EmployeeTable  
empNo, empName, empEmail, EmpGrade



EmployeeProjectTable  
empNo, proNo, hrsWorked  
  


1. **Wellmeadows Hospital Patient Medication Normalization.**

**(a) Functional Dependencies:**

1. **Patient Number -> FullName, Bed Number, Ward Number, Ward Name (FD1):** This assumes that the patient number is assigned to each patient is unique and it uniquely identifies all the other information for a patient.
2. **Drug Number -> Name, Description, Dosage, Method of Admin, Units per Day:** The drug number uniquely identifies a specific medication, and its details depend solely on that identifier.
3. **(Drug Number, Start Date, Finish Date) -> Name, Description, Dosage, Method of Admin, Units per Day:** This assumes that a unique combination of drug number, start date, and finish date identifies the specific medication given to a patient.

**Assumptions:**

* All attributes are not null (they must have a value).
* The drug number is also unique within the hospital system.

**(b) Normalization to 3NF:**

The current form shown in figure 14.18 violates 3NF due to a transitive dependency.

**1. Decompose into two relations:**

* **Patient:**
  + Patient Number (Primary Key)
  + FullName
  + Bed Number
  + Ward Number
  + Ward Name
* **Medication:**
  + Drug Number (Primary Key)
  + Patient Number (Foreign Key)
  + Name
  + Description
  + Dosage
  + Method of Admin
  + Units per Day
  + StartDate
  + FinishDate
* **2.** By separating the data into two tables:
  + **Patient:** Stores patient information that is independent of specific medication administration.
  + **Medication:** Stores medication details and administration details, linked to the patient through the **Patient Number** foreign key.

This normalization removes redundancy, simplifies data manipulation, and enhances data integrity.

**(c) Keys:**

**Patient:**

* Primary Key: Patient Number

**Medication:**

* Primary Key: Drug Number
* Foreign Key: Patient Number references the patient table ensuring data integrity.

1. **Dentist/Patient appointment Normalization.**

**a). Update Anomalies:**

The update anomalies are described as follows: -

**Insertion Anomaly:**

* If a new dentist needs to be added, but they haven't yet had any appointments, it's impossible to insert their record without violating the table's structure, as there would be no associated patient or appointment details.

**Deletion Anomaly:**

* If a patient cancels all their appointments, their record might be deleted, losing information about them.

**Update Anomaly:**

* If a dentist's staff number changes, it would need to be updated in every row where the dentist appears, potentially causing inconsistencies if the update is not done consistently.

**b). Functional Dependencies :**

Assumptions:

* StaffNo uniquely identifies a dentist.
* patNo uniquely identifies a patient.
* SugeryNo uniquely identifies a surgery.

The Functional Dependencies are:

* StaffNo -> dentistName (determines the dentist's name)
* patNo -> patName (determines the patient's name)
* (StaffNo, appointment date time) -> SugeryNo (combines dentist + appointment time to determine assigned surgery)

**c). Normalization to 3NF:**

To eliminate redundancy and ensure data integrity, we need to decompose the table into three separate relations:

* **Dentist:**
  + StaffNo (Primary Key, uniquely identifies a dentist)
  + DentistName
* **Patient:**
  + patNo (Primary Key, unique patient identifier)
  + patName
* **Appointment:**
  + StaffNo (Foreign Key, references Dentist)
  + patNo (Foreign Key, references Patient)
  + appointmentDateTime
  + SugeryNo (Foreign Key, references Sugery)

**Details about the tables above:**

* The original table combined information about dentists, patients, and appointments into a single entity.
* By separating the data into three tables:
  + **Dentist:** Captures dentist details, independent of specific appointments.
  + **Patient:** Stores patient information, independent of appointments.
  + **Appointment:** Links dentists, patients, and appointment details, referencing the other two tables through foreign keys.

**Keys:**

* **Dentist:**
  + Primary Key: StaffNo (uniquely identifies a dentist)
* **Patient:**
  + Primary Key: patNo (uniquely identifies a patient)
* **Appointment:**
  + Primary Key: (composite key) (StaffNo, appointmentDateTime) (combination of dentist and appointment time uniquely identifies an appointment)
  + Foreign Key: StaffNo (references Dentist.StaffNo)
  + Foreign Key: patNo (references Patient.patNo)