Data Store Design

& Implementation

NAME:	LYAN HENG
ID:	102240691
DATE OF SUBMISSION:	30/10/2020
CLASS:	COS20015 –

FUNDAMENTAL OF DATA MANAGEMENT

Table of Content

I. Report Description

This report goes over the process of designing and implementing a custom made database. The process involves studying the scenario, designing the database, improving the design through various methods and aspects, building the database, and inserting valid data. Each step of the process is explained and demonstrated in detail to provide evidence of the understanding of the task outcome alignments of the D task. It also closely aligns to the unit's overall learning outcomes.

Task Outcome Alignment:

- Storage Efficiency : correctly designed and implemented an efficient custom database.
- Schemas and Validation: correctly designed and demonstrated the correctness of all the queries and design made.
- Relational Principles: applied the understanding of relational databases and understanding of normalisation to improve the design of the database.
- Data Access Tools: Demonstrated the effective uses of MySQL on XAMPPs and MySQL Workbench. Correctly and efficiently applied the use of external tools like research and learning materials from the internet to improve the design further.

Learning Outcome:

- Appreciate the algorithmic complexity in the context of data management.
- Use techniques, tools, and methods to sort, search, and transform data stored.
- Explain the role of data types, data representation, schemas in managing data.
- Use appropriate methods to efficiently store, insert, and retrieve data appreciating the underlying trades-offs between different strategies.
- Understanding basic concurrency and addressing those issues.

II. Overall Description - Scenario

The design is based on a database design for Softtech, a new startup software development company. The company runs on a project basis. The following are the details received from the company:

- Each project requires a **project manager**. A project manager is identified by an ID number. Each project manager also has a name, a gender, an age, an address, an email, and a phone number. Each project manager can manage one or more projects.

- Each **project** represents the details there is available for a project. Each project is identified by an ID and includes details like start date, end date, location, project manager in charge, a bill number, status of completion, and comments. Projects can also be divided into three different categories: small, medium, and large. The number of developers there can categorize projects needs to be for the project (1: small, 2-4: medium, 5-above: large) and the estimated income of the project.

- The company also requires each **project's progress** to be entered into the database. It will be used to record any actions taken in the project. Each project progress' entry will also need to include the developers that are responsible for certain actions.

- **Developer**'s information will also be recorded. Each developer has all the personal information the project manager has.

- All the developer's and project manager's current **salary** will also be recorded. Since the team is quite small, many of the developers will be working in many different projects.

- **Clients**' information will also be recorded; including their name (company/individual), representative's name, phone number, email, date of start of business, and the number of projects requested.

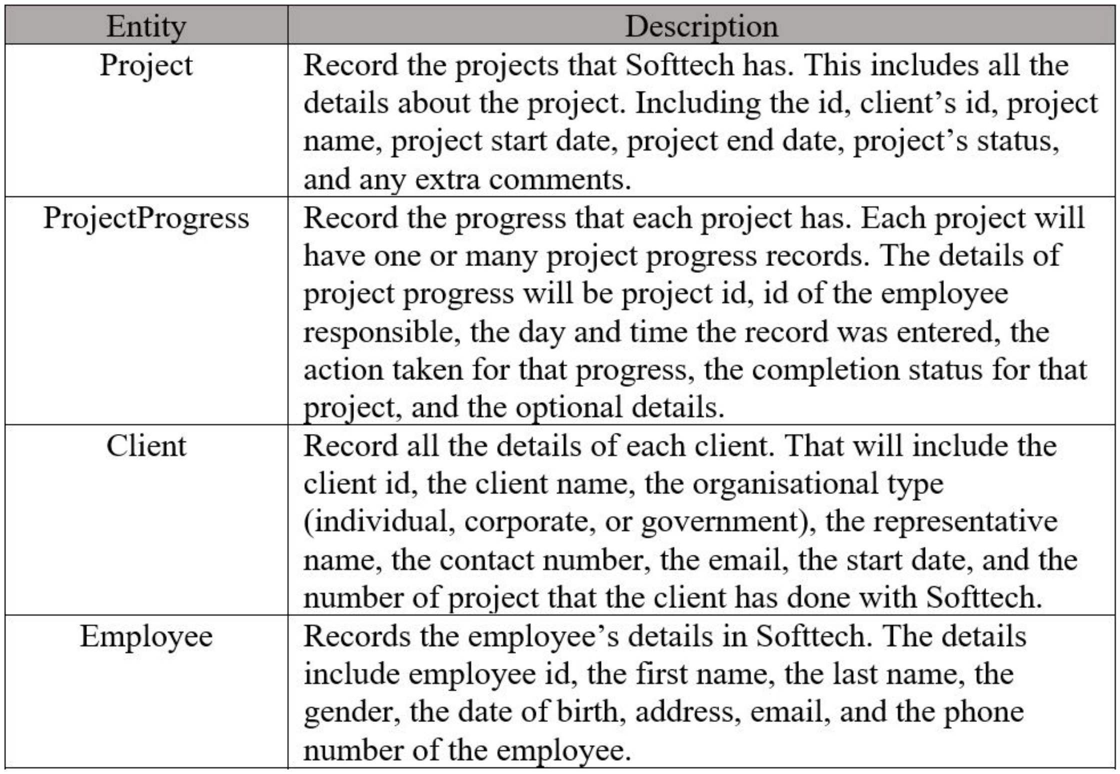
- The founder of SoftTech also emphasizes the importance of recording a separate entry of all **bills** that occur.

- A **final year report of bills** is also required. The bill will show how much the company earns from each client. Since the company is a small startup, all bills are checked and verified by only one accountant.

- The founder also appreciates some **data-automation** or features that can improve the company's work efficiency and effectiveness.

III. Creation of Entities (Tables)

Description of Each Table:



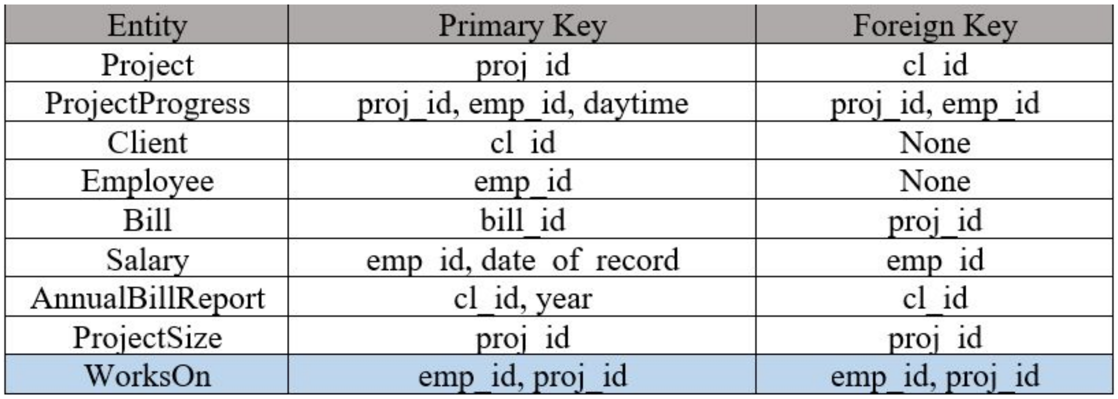
Entity	Description
Project	Record the projects that Softtech has. This includes all the details about the project. Including the id, client's id, project name, project start date, project end date, project's status, and any extra comments.
ProjectProgress	Record the progress that each project has. Each project will have one or many project progress records. The details of project progress will be project id, id of the employee responsible, the day and time the record was entered, the action taken for that progress, the completion status for that project, and the optional details.
Client	Record all the details of each client. That will include the client id, the client name, the organisational type (individual, corporate, or government), the representative name, the contact number, the email, the start date, and the number of project that the client has done with Softtech.
Employee	Records the employee's details in Softtech. The details include employee id, the first name, the last name, the gender, the date of birth, address, email, and the phone number of the employee.

Employee	Records the employee's details in Softtech. The details include employee id, the first name, the last name, the gender, the date of birth, address, email, and the phone number of the employee.
Bill	Records the bills that the company has issued. The details includes the bill id, the project id, the day and time it was recorded, the work done for that specific bill, the hours the work was done, the hourly rate of the bill, the total bill, the status of the bill ('completed','issued', 'ready to issue', 'not ready')
Salary	Records the salary of each employee each month. The details of this includes the employee id, the date of record, the base salary, the hours the employee put in in the month, the bonus the employee receives in the month, and the total the employee is paid.
AnnualBillReport	Records the yearly bill report. This is updated yearly. The use of this is to show how much the company earns in each year. The detail of this includes the client id, the year of record, the income, and the number of projects done in the years.
ProjectSize	Records the sizes of the projects, ranking them by size and importance. The details of this includes the project id, the size of the project, the importance, the number of developers, and extra comments. This has a one to one relationship with the Project table.
WorksOn	This is a weak entity made to eliminate the many to many relationship between employees and projects. This will record the employee's id and the project id that corresponds to the the ones they work on. It will also include the information of what role the employee takes in the project, and any extra comments if necessary.

Initial Design of Existing Entities (Tables): (Including Many-Many Relationship)

Entity	Primary Key	Foreign Key
Project	proj_id	cl_id, emp_id
ProjectProgress	proj_id, emp_id, daytime	proj_id, emp_id
Client	cl_id	None
Employee	emp_id	proj_id
Bill	bill_id	proj_id
Salary	emp_id, date_of_record	emp_id
AnnualBillReport	cl_id, year	cl_id
ProjectSize	proj_id	proj_id

Due to the many-to-many relationship issue explained below (Cardinality Section), another entity is added. The final entity table should look like this:



Entity	Primary Key	Foreign Key
Project	proj_id	cl_id
ProjectProgress	proj_id, emp_id, daytime	proj_id, emp_id
Client	cl_id	None
Employee	emp_id	None
Bill	bill_id	proj_id
Salary	emp_id, date_of_record	emp_id
AnnualBillReport	cl_id, year	cl_id
ProjectSize	proj_id	proj_id
WorksOn	emp_id, proj_id	emp_id, proj_id

IV. Cardinality

According to DatabaseZone, many-to-many relationships are not acceptable to be implemented, as it is very prone to anomalies (Brumm 2017). A concept called the joining table or bridging table was used to fix up any table with a many-to-many relationship. In this case, there is one many-to-many relationship, which is between Project and Employee. Each project has many employees working on it, and each employee can work on multiple projects (up to 3) as well. To eliminate this many-to-many relationship, we will be adding a new entity called 'Works On'. This will have information about employees that are currently working on a related project. It will also include the employee's role in the project.

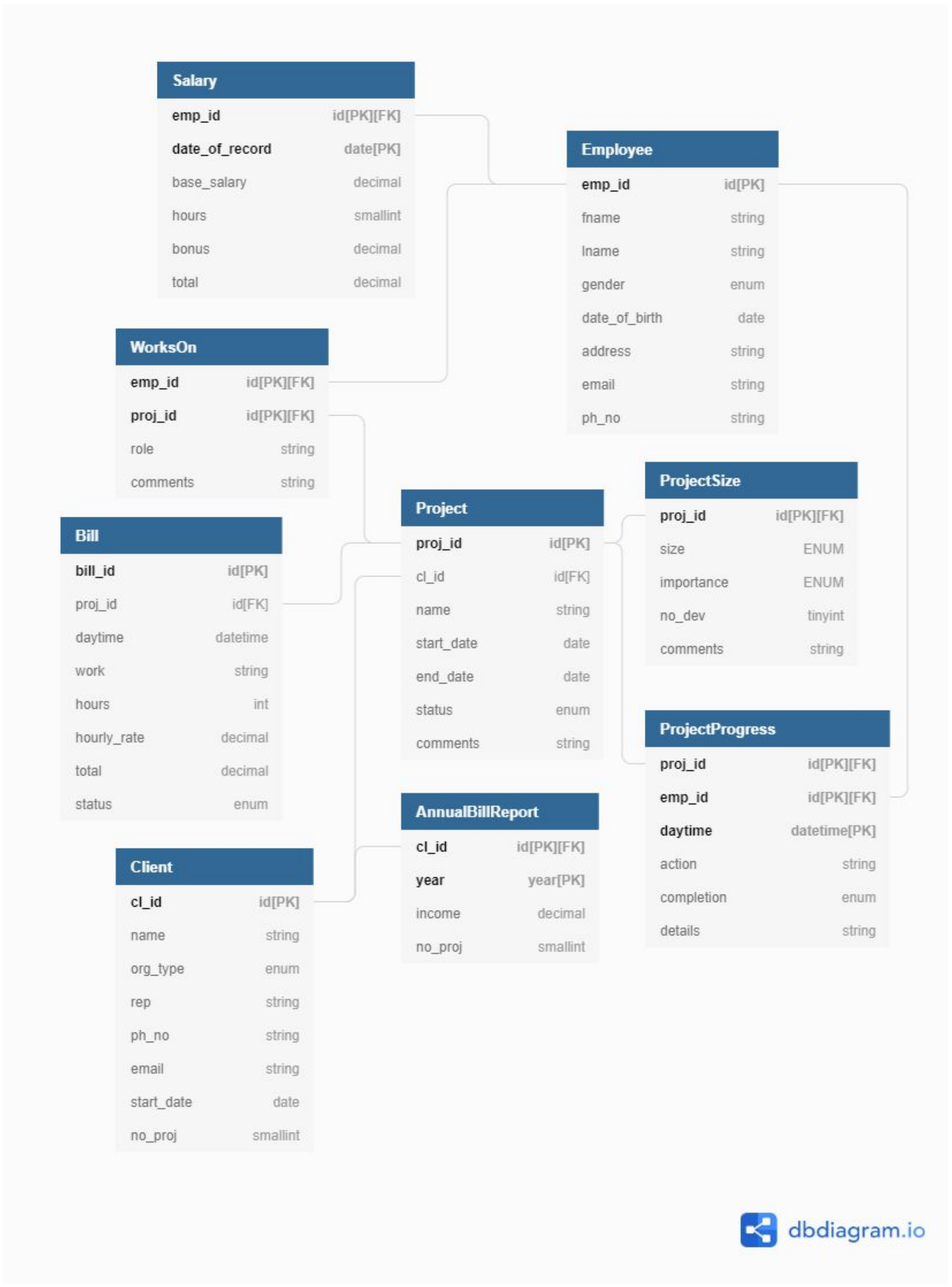
Original cardinality:

Entity	Relationship	Entity
Project	one-to-many	Bill
Project	one-to-many	ProjectProgress
Project	many-to-many	Employee
Client	one-to-many	Project
Employee	one-to-one	Salary
Employee	one-to-many	Project Progress
ProjectSize	one-to-one	Project
BillReport	one-to-one	Client

The final cardinality table should look like this:

Entity	Relationship	Entity
Project	one-to-many	Bill
Project	one-to-many	ProjectProgress
Project	one-to-many	WorksOn
Employee	one-to-many	WorksOn
Client	one-to-many	Project
Employee	one-to-one	Salary
Employee	one-to-many	Project Progress
ProjectSize	one-to-one	Project
BillReport	one-to-one	Client

V. UML Diagram



A full UML Diagram (with cardinality) can be found in the link attached: https://dbdiagram.io/d/5f54691e88d052352cb61a2e

VI. Normalisation Process

Normalization is an essential aspect of a database. An extremely denormalized database presents many chances for data inconsistency. A too normalized and over-engineered database can also be complicated to manage. Finding the right normalized form for each specific situation is essential to ensure the balance between data security and performance.

A well-designed real-world database will be in the 2nd or the 3rd Normal Form. In the 2nd Normal Form, partial dependencies are eliminated. In the 3rd Normal Form, transitive dependencies are eliminated. Even though most databases are in the 3rd Normal Form, according to LifeWire, it is not an absolute requirement. Sometimes, the design can even be improved by violating the 3rd Normal Form (Chapple 2020). Softtech's database will be in the 2nd and 3rd Normal Form.

Project

proj_id < PK >	cl_id < FK >	name	start_date	end_date	status	comments

Every element in the table depends on the PK and can be determined by the PK. There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

ProjectProgress

proj_id < PK FK>	emp_id <PK FK>	daytime <FK>	action	completion	details

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

Client

cl_id < PK >	name	org_type	rep	ph_no	email	start_date	no_proj

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

Employee

emp_id < PK >	fname	lname	gender	date_of_ birth	address	email	ph_no

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

Bill

bill_id < PK >	proj_id < FK >	daytime	work	hours	hourly_ rate	total	status

Almost all of the elements are functionally dependent on the PK. The element 'total' can be calculated and inferred from hours and hourly_rate. This presents an occurrence of transitive dependency. However, since 'total' can always be programmed to be the multiplication of hours and hourly_rate, it is guaranteed that there will not be inconsistencies in the database. It would also be inefficient to have a separate table to calculate the 'total' bill for each bill. This table is in the 2nd Normal Form.

Salary

emp_id < PK FK >	Date_of_ record < PK >	base_salary	hours	bonus	total

This is similar to the Bill table. The element 'total' only depends on 'base_salary', 'hours', and 'bonus'. The three elements are then dependent on the PK. This is another transitive dependency that can be programmed never to have data inconsistency. This table is also in the 2nd Normal Form.

WorksOn

emp_id < PK FK>	proj_id <PK FK>	role	comments

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

AnnualBillReport

cl_id < PK FK>	year < PK >	income	no_proj

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

ProjectSize

proj_id < PK FK >	size	importance	no_dev	comments

Every non-primary attribute depends on the primary key (all of the candidate keys). There are no transitive nor partial dependencies. This table is in the 3rd Normal Form.

Summary:

In general, the tables are in the 3rd Normal Form. However, as a database, it is in its 2nd Normal Form, due to some tables being in the 2nd Normal Form. This database is designed to eliminate most of data inconsistencies that might be caused by humans or by the database manager itself. In the Data Insertion & Constraint section, there are also triggers that are designed to check the correctness of data insertion.

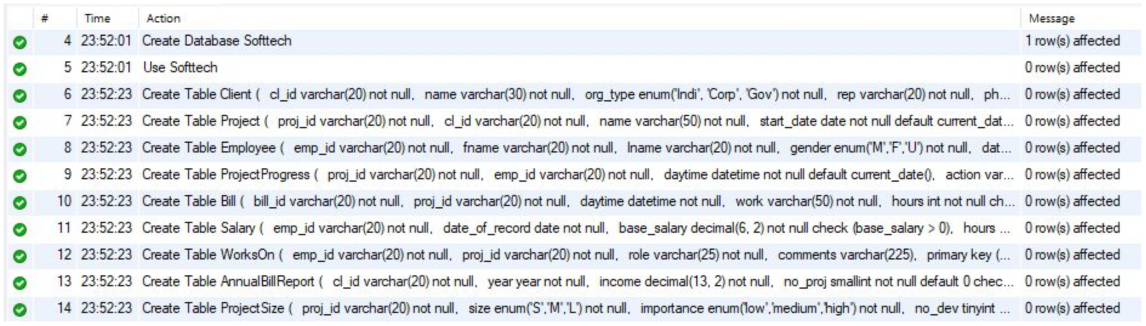
VII. Table Creation & Constraints

Before the data can be inserted, the database has to be built. The query below will build the needed tables.

```
-- Creating the tables
-- Client table
Create Table Client (
  cl_id varchar(20) not null,
  name varchar(30) not null,
  org_type enum('Indi', 'Corp', 'Gov') not null,
  rep varchar(20) not null,
  ph_no varchar(15) not null,
  email varchar(30) not null,
  start_date date not null default current_date(),
  no_proj smallint not null default 0,
  primary key (cl_id)
);

-- Project table
Create Table Project (
  proj_id varchar(20) not null,
  cl_id varchar(20) not null,
  name varchar(50) not null,
  start_date date not null default current_date() ,
  end_date date,
  status enum('completed', 'pending', 'not started') not null default
"not started",
  comments varchar(225),
```

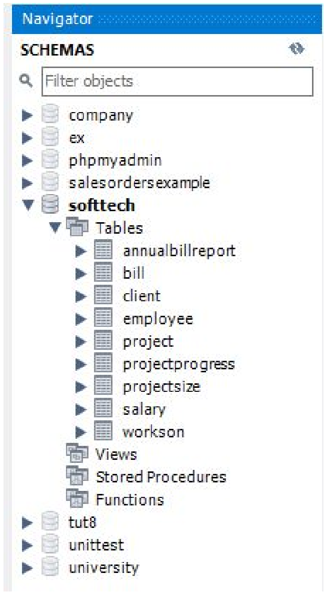
```sql
  primary key (proj_id),
  foreign key (cl_id) references Client(cl_id)
);

-- Employee table
Create Table Employee (
  emp_id varchar(20) not null,
  fname varchar(20) not null,
  lname varchar(20) not null,
  gender enum('M','F','U') not null,
  date_of_birth date not null,
  start_date date not null,
  address varchar(150) not null,
  email varchar(30) not null,
  ph_no varchar(15) not null,
  primary key (emp_id)
);

-- ProjectProgress table
Create Table ProjectProgress (
  proj_id varchar(20) not null,
  emp_id varchar(20) not null,
  daytime datetime not null default current_date(),
  action varchar(225) not null,
  completion enum('completed', 'on-progress', 'not started') not null,
  details varchar(225) not null,
  primary key (proj_id, emp_id, daytime),
  foreign key (proj_id) references Project(proj_id),
  foreign key (emp_id) references Employee(emp_id)
);

-- Bill table
Create Table Bill (
  bill_id varchar(20) not null,
  proj_id varchar(20) not null,
  daytime datetime not null,
  work varchar(50) not null,
  hours int not null check (hours >= 0),
  hourly_rate decimal(7, 2) not null check (hours >= 0),
  total decimal(9, 2) generated always as (hours * hourly_rate),
  status enum('completed','issued', 'ready to issue', 'not ready') not
null,
  primary key (bill_id),
  foreign key (proj_id) references Project(proj_id)
);

Create Table Salary (
  emp_id varchar(20) not null,
  date_of_record date not null,
  base_salary decimal(6, 2) not null check (base_salary > 0),
  hours smallint not null check (hours > 0),
  bonus decimal(5, 2) not null default 0.00 check (bonus >= 0),
  total decimal(7, 2) generated always as (base_salary * hours + bonus),
  primary key (emp_id, date_of_record),
  foreign key (emp_id) references Employee(emp_id)
);

-- WorksOn table
Create Table WorksOn (
  emp_id varchar(20) not null,
  proj_id varchar(20) not null,
  role varchar(25) not null,
  comments varchar(225),
  primary key (proj_id, emp_id),
  foreign key (proj_id) references Project(proj_id),
  foreign key (emp_id) references Employee(emp_id)
);

-- AnnualBillReport table
Create Table AnnualBillReport (
  cl_id varchar(20) not null,
  year year not null,
  income decimal(13, 2) not null,
  no_proj smallint not null default 0 check (no_proj > 0),
  primary key (cl_id, year),
  foreign key (cl_id) references Client(cl_id)
);

-- ProjectSize table
Create Table ProjectSize (
  proj_id varchar(20) not null,
  size enum('S','M','L') not null,
  importance enum('low','medium','high') not null,
  no_dev tinyint not null check (no_dev > 0),
  comments varchar(225),
  primary key (proj_id),
  foreign key (proj_id) references Project(proj_id)
);
```

Query ran:



```
#   Time      Action                                                                                                                                                                                Message
4   23:52:01  Create Database Softtech                                                                                                                                                              1 row(s) affected
5   23:52:01  Use Softtech                                                                                                                                                                          0 row(s) affected
6   23:52:23  Create Table Client (  cl_id varchar(20) not null,  name varchar(30) not null,  org_type enum('Indi', 'Corp', 'Gov') not null,  rep varchar(20) not null,  ph...                      0 row(s) affected
7   23:52:23  Create Table Project (  proj_id varchar(20) not null,  cl_id varchar(20) not null,  name varchar(50) not null,  start_date date not null default current_dat...                          0 row(s) affected
8   23:52:23  Create Table Employee (  emp_id varchar(20) not null,  fname varchar(20) not null,  lname varchar(20) not null,  gender enum('M','F','U') not null,  dat...                           0 row(s) affected
9   23:52:23  Create Table ProjectProgress (  proj_id varchar(20) not null,  emp_id varchar(20) not null,  daytime datetime not null default current_date(),  action var...                      0 row(s) affected
10  23:52:23  Create Table Bill (  bill_id varchar(20) not null,  proj_id varchar(20) not null,  daytime datetime not null,  work varchar(50) not null,  hours int not null ch...                    0 row(s) affected
11  23:52:23  Create Table Salary (  emp_id varchar(20) not null,  date_of_record date not null,  base_salary decimal(6, 2) not null check (base_salary > 0),  hours ...                         0 row(s) affected
12  23:52:23  Create Table WorksOn (  emp_id varchar(20) not null,  proj_id varchar(20) not null,  role varchar(25) not null,  comments varchar(225),  primary key (...                            0 row(s) affected
13  23:52:23  Create Table AnnualBillReport (  cl_id varchar(20) not null,  year year not null,  income decimal(13, 2) not null,  no_proj smallint not null default 0 chec...                      0 row(s) affected
14  23:52:23  Create Table ProjectSize (  proj_id varchar(20) not null,  size enum('S','M','L') not null,  importance enum('low','medium','high') not null,  no_dev tinyint ...                      0 row(s) affected
```

<u>Table Exists:</u>



```
Navigator
SCHEMAS
Filter objects
▶ company
▶ ex
▶ phpmyadmin
▶ salesordersexample
▼ softtech
    ▼ Tables
        ▶ annualbillreport
        ▶ bill
        ▶ client
        ▶ employee
        ▶ project
        ▶ projectprogress
        ▶ projectsize
        ▶ salary
        ▶ workson
      Views
      Stored Procedures
      Functions
▶ tut8
▶ unittest
▶ university
```

Constraints include:

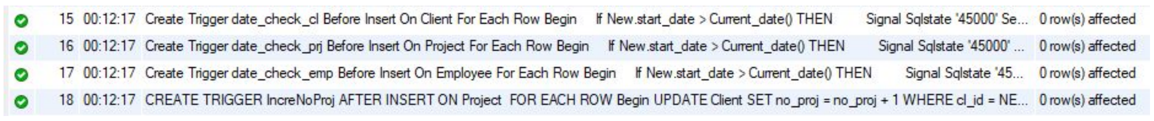
- primary and foreign key constraints.
- It also includes default, not null, and check constraints.
- Checking that the age of the worker is above 18
- Check that the hour of work is above 0.
- Make sure that Softtech's employee's mistake during data entry will be alerted and taken care of.
- However, with the use of MySQL version 7.0 and above, many other check constraints cannot be done in the table creation queries. This is due to check constraints not allowing unpredictable constants, when current_date() is an unpredictable constant. Another solution is made to compensate for this. The check constraints can be set up using a trigger clause.

Trigger Queries:

```
-- Create triggers to check that start_date is not in the future
-- Client - start_date
-- Checking that the client's start_date with softtech is not before
today
DELIMITER //
Create Trigger date_check_cl
Before Insert
On Client For Each Row
Begin
    If New.start_date > Current_date() THEN
        Signal Sqlstate '45000' Set Message_text = 'Invalid date';
    End If;
End; //

-- Project - start_date
-- Checking that the project start_date is not in the future
-- Each employee are required to enter the project description into the
DB when the DB is confirmed
DELIMITER //
Create Trigger date_check_prj
Before Insert
On Project For Each Row
Begin
    If New.start_date > Current_date() THEN
        Signal Sqlstate '45000' Set Message_text = 'Invalid date';
    End If;
End; //

-- Employee - start_date
-- Checking that employee are not entered in advanced
-- Confirmation is needed every time an employee is entered into the DB
DELIMITER //
Create Trigger date_check_emp
Before Insert
On Employee For Each Row
Begin
    If New.start_date > Current_date() THEN
        Signal Sqlstate '45000' Set Message_text = 'Invalid date';
    End If;
End; //

-- increment Client's no_proj when Project table is inserted
-- This is to make sure that the client's no_proj is updated
-- everytime a new project is inserted into the DB.
-- Implementation of an auto-increment of a non-PK
CREATE TRIGGER IncreNoProj AFTER INSERT ON Project
FOR EACH ROW
Begin
    UPDATE Client
    SET no_proj = no_proj + 1
    WHERE cl_id = NEW.cl_id;
End; //
```

The triggers make sure that after every insertion, if any issue is detected, it will be alerted. There is also an introduction of the last trigger, where it updates the Client's 'no_proj' attribute every time a new project is added into the Project table. This makes sure the database does not have any data inconsistency or lost updates.

Query ran:



```
✓ 15 00:12:17 Create Trigger date_check_cl Before Insert On Client For Each Row Begin   If New.start_date > Current_date() THEN   Signal Sqlstate '45000' Se...   0 row(s) affected
✓ 16 00:12:17 Create Trigger date_check_prj Before Insert On Project For Each Row Begin   If New.start_date > Current_date() THEN   Signal Sqlstate '45000' ...   0 row(s) affected
✓ 17 00:12:17 Create Trigger date_check_emp Before Insert On Employee For Each Row Begin   If New.start_date > Current_date() THEN   Signal Sqlstate '45...   0 row(s) affected
✓ 18 00:12:17 CREATE TRIGGER IncreNoProj AFTER INSERT ON Project  FOR EACH ROW Begin UPDATE Client SET no_proj = no_proj + 1 WHERE cl_id = NE...   0 row(s) affected
```

VIII. Data Insertion & Automation

After setting up the tables and constraints, the data is inserted into the database with the insert clauses, as given below.

Insert Queries:

```
-- Adding records into the database (minimum 25 records & 5 JOINs)
-- Insert into Employees
Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ("EMP001", "Lyan", "Heng", 'M', '1995-04-19', '2019-03-19', "32
Jones Road WESTLAKE, Queensland(QLD), 4074", "lyanh@softtech.com",
"+61403193849");

Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ("EMP002", "John", "Wit", 'M', "1999-12-07", "2019-04-03", "95
Zipfs Road WEST IPSWICH, Queensland(QLD), 4305", "johnw@softtech.com",
"+61401820391");

Insert Into Employee(emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ("EMP003", "Jim", "Tim", 'M', "1997-10-04", "2019-05-01", "42 Sale
Street LIDSTER, New South Wales(NSW), 2800", "jimt@softtech.com",
"+61419203920");

Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ("EMP004", "Ashley", "Nguyen", 'F', "2001-02-01", "2019-07-29",
"15 Tennyson Road LAKEMBA, New South Wales(NSW), 2195",
"ashleyn@softtech.com", "+61401923019");
```

```
Insert Into Employee (emp_id, fname, lname, gender, date_of_birth,
start_date, address, email, ph_no)
Values ("EMP005", "Natalie", "Wilkinson", 'U', "2001-02-01",
"2020-01-09", "41 Shirley Street HOLMVIEW, Queensland(QLD), 4207",
"nataliew@softtech.com", "+61412938210");

-- Insert into Client
Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date)
Values ("CLI001", "Jay Tim Technology Co. Ltd", "Corp", "May Dimson",
"0347538293", "buspartner@jaytim.com","2019-02-05");

Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date)
Values ("CLI002", "MacroHard Technology Co. Ltd", "Corp", "Tom Polland",
"03419230291", "contact@mt.com", "2019-12-03");

Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date)
Values ("CLI003", "Australian Taxation Office", "Gov", "Jim Carter",
"03029103192", "jc@ato.au.gov", "2019-02-05");

-- Insert into Project
Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
Values ("PRJ001", "CLI001", "Jay Tim's Database System", "2019-02-05",
"2019-08-04", "completed", "Create a database system for Jay Tim.");

Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
Values ("PRJ002", "CLI002", "Microcontroller Design", "2019-05-02",
"2019-08-01", "completed", "Microcontroller for Door System");

Insert Into Project (proj_id, cl_id, name, start_date, end_date, status,
comments)
Values ("PRJ003", "CLI003", "Website Design", "2019-07-02", "2019-08-10",
"completed", "Small section of website for ATO");

Insert Into Project (proj_id, cl_id, name, start_date, status, comments)
Values ("PRJ004", "CLI002", "Door Safety System", "2019-08-02",
"on-progress", "A sequel after the door micro-design");

-- Insert into ProjectProgress
Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ001", "EMP001", "2019-03-10 11:45:03", "Completed Design",
"completed", "After a long study of company structure, design is
completed.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
```

```
Values ("PRJ002", "EMP003", "2019-05-02 08:34:53", "Project started",
"completed", "Contract Signed. Project proceeded right away.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ001", "EMP002", "2019-05-22 18:07:37", "Database Draft
Completed", "completed", "A draft of the database completed. Proceed to
testing");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ002", "EMP003", "2019-06-01 08:12:34", "Microcontroller
Finalisation", "pending", "Report on finalisation's progress.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ003", "EMP001", "2019-07-02 12:30:22", "Project started",
"completed", "Contract signed. Project started to be built.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ002", "EMP004", "2019-08-01 08:00:00", "Project finalised",
"completed", "Project has been finalised.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ004", "EMP003", "2019-08-02 08:32:12", "Project started",
"completed", "Contract signed.");

Insert Into ProjectProgress (proj_id, emp_id, daytime, action,
completion, details)
Values ("PRJ003", "EMP001", "2019-08-10 12:30:00", "Project completed",
"completed", "Website built to satisfaction");

-- Insert into WorksOn
Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP001", "PRJ001", "Project Manager", "Responsible for the
progress and completion of the project");

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP002", "PRJ001", "Developer", "Develop the database system");

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP003", "PRJ002", "Project Manager", "Responsible for the
progress and completion of the project");

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP004", "PRJ002", "Developer", "Microcontroller Designer");
```

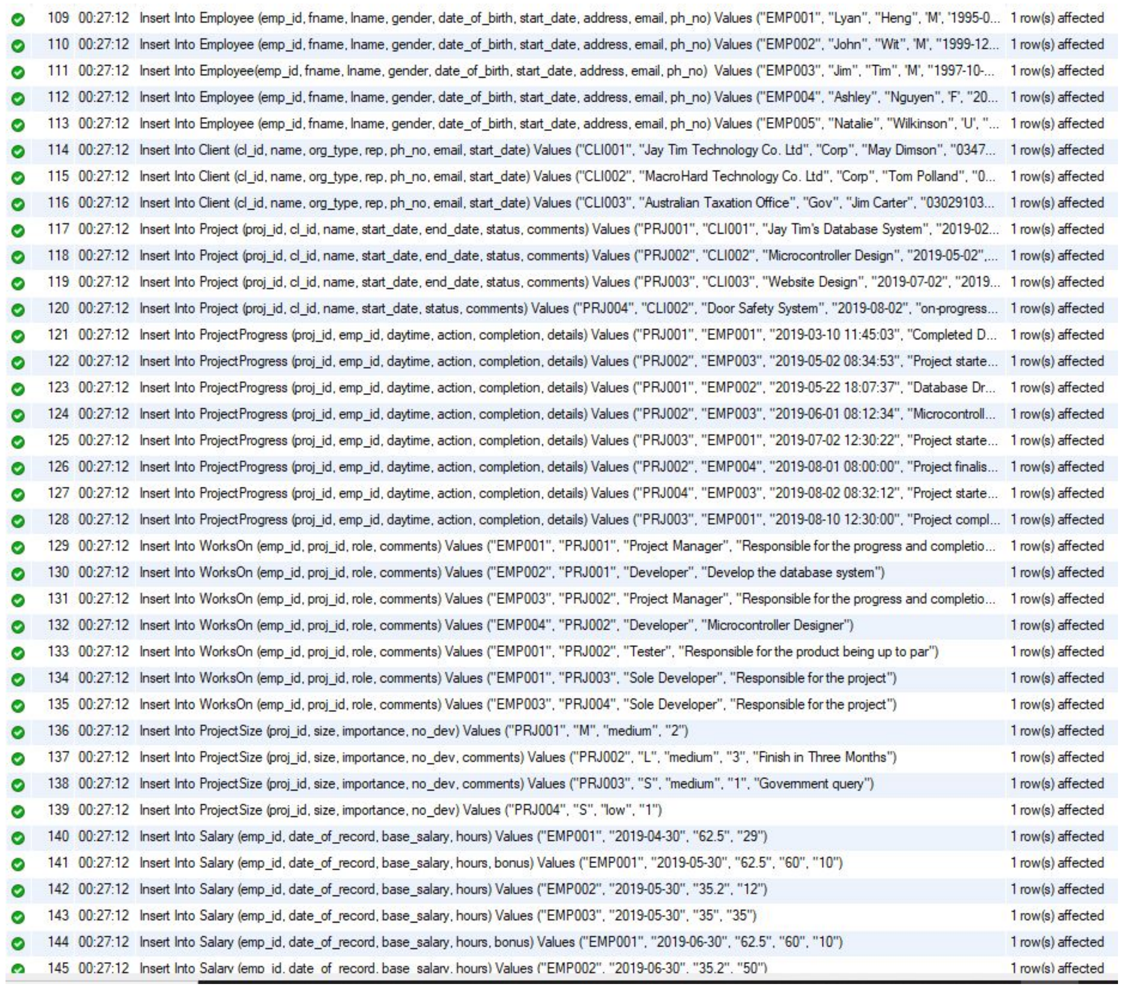
```sql
Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP001", "PRJ002", "Tester", "Responsible for the product being
up to par");

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP001", "PRJ003", "Sole Developer", "Responsible for the
project");

Insert Into WorksOn (emp_id, proj_id, role, comments)
Values ("EMP003", "PRJ004", "Sole Developer", "Responsible for the
project");

-- Insert into ProjectSize
Insert Into ProjectSize (proj_id, size, importance, no_dev)
Values ("PRJ001", "M", "medium", "2");

Insert Into ProjectSize (proj_id, size, importance, no_dev, comments)
Values ("PRJ002", "L", "medium", "3", "Finish in Three Months");

Insert Into ProjectSize (proj_id, size, importance, no_dev, comments)
Values ("PRJ003", "S", "medium", "1", "Government query");

Insert Into ProjectSize (proj_id, size, importance, no_dev)
Values ("PRJ004", "S", "low", "1");

-- Insert into Salary
Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP001", "2019-04-30", "62.5", "29");

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ("EMP001", "2019-05-30", "62.5", "60", "10");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-05-30", "35.2", "12");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-05-30", "35", "35");

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ("EMP001", "2019-06-30", "62.5", "60", "10");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-06-30", "35.2", "50");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-06-30", "35", "30");

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ("EMP001", "2019-07-30", "62.5", "60", "20");
```

```
Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-07-30", "35.2", "45");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-07-30", "35", "35");

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ("EMP001", "2019-08-30", "62.5", "58", "5");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-08-30", "35.2", "50");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-08-30", "35", "35");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2019-08-30", "40", "5");

Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus)
Values ("EMP001", "2019-09-30", "62.5", "60", "10");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-09-30", "35.2", "45");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-09-30", "35", "20");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2019-09-30", "40", "25");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP001", "2019-10-30", "62.5", "62");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-10-30", "35.2", "55");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-10-30", "35", "20");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2019-10-30", "40", "15");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP001", "2019-11-30", "62.5", "62");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-11-30", "35.2", "60");
```

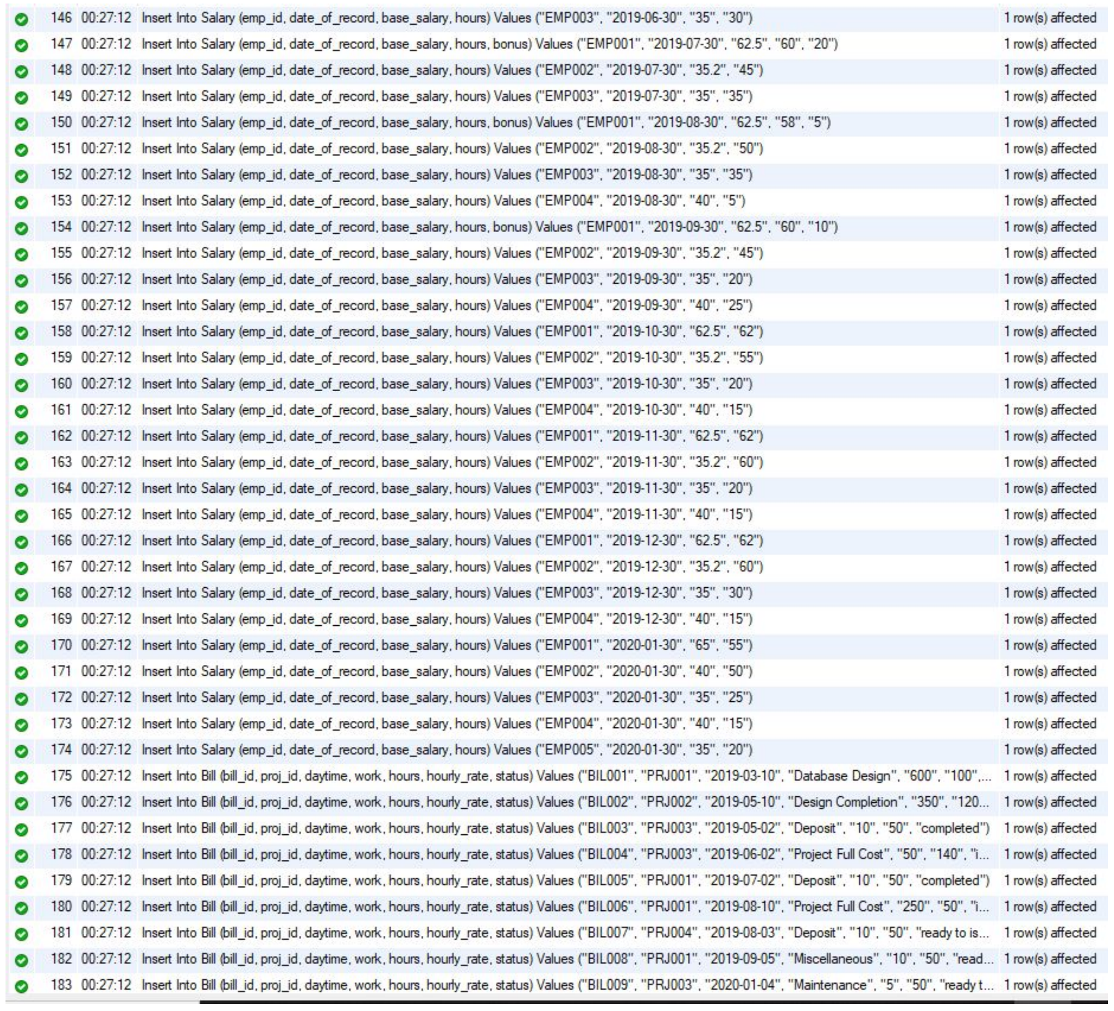
```
Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-11-30", "35", "20");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2019-11-30", "40", "15");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP001", "2019-12-30", "62.5", "62");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2019-12-30", "35.2", "60");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2019-12-30", "35", "30");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2019-12-30", "40", "15");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP001", "2020-01-30", "65", "55");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP002", "2020-01-30", "40", "50");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP003", "2020-01-30", "35", "25");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP004", "2020-01-30", "40", "15");

Insert Into Salary (emp_id, date_of_record, base_salary, hours)
Values ("EMP005", "2020-01-30", "35", "20");

-- Insert into Bill
Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL001", "PRJ001", "2019-03-10", "Database Design", "600",
"100", "completed");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL002", "PRJ002", "2019-05-10", "Design Completion", "350",
"120", "completed");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL003", "PRJ003", "2019-05-02", "Deposit", "10", "50",
"completed");
```

```
Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL004", "PRJ003", "2019-06-02", "Project Full Cost", "50",
"140", "issued");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL005", "PRJ001", "2019-07-02", "Deposit", "10", "50",
"completed");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL006", "PRJ001", "2019-08-10", "Project Full Cost", "250",
"50", "issued");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL007", "PRJ004", "2019-08-03", "Deposit", "10", "50", "ready
to issue");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL008", "PRJ001", "2019-09-05", "Miscellaneous", "10", "50",
"ready to issue");

Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate,
status)
Values ("BIL009", "PRJ003", "2020-01-04", "Maintenance", "5", "50",
"ready to issue");
```

Queries ran:



```
109 00:27:12 Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_date, address, email, ph_no) Values ("EMP001", "Lyan", "Heng", 'M', '1995-0...   1 row(s) affected
110 00:27:12 Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_date, address, email, ph_no) Values ("EMP002", "John", "Wit", 'M', "1999-12...   1 row(s) affected
111 00:27:12 Insert Into Employee(emp_id, fname, lname, gender, date_of_birth, start_date, address, email, ph_no)  Values ("EMP003", "Jim", "Tim", 'M', "1997-10-...   1 row(s) affected
112 00:27:12 Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_date, address, email, ph_no) Values ("EMP004", "Ashley", "Nguyen", 'F', "20...   1 row(s) affected
113 00:27:12 Insert Into Employee (emp_id, fname, lname, gender, date_of_birth, start_date, address, email, ph_no) Values ("EMP005", "Natalie", "Wilkinson", 'U', "...   1 row(s) affected
114 00:27:12 Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Values ("CLI001", "Jay Tim Technology Co. Ltd", "Corp", "May Dimson", "0347...   1 row(s) affected
115 00:27:12 Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Values ("CLI002", "MacroHard Technology Co. Ltd", "Corp", "Tom Polland", "0...   1 row(s) affected
116 00:27:12 Insert Into Client (cl_id, name, org_type, rep, ph_no, email, start_date) Values ("CLI003", "Australian Taxation Office", "Gov", "Jim Carter", "03029103...   1 row(s) affected
117 00:27:12 Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, comments) Values ("PRJ001", "CLI001", "Jay Tim's Database System", "2019-02...   1 row(s) affected
118 00:27:12 Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, comments) Values ("PRJ002", "CLI002", "Microcontroller Design", "2019-05-02",...   1 row(s) affected
119 00:27:12 Insert Into Project (proj_id, cl_id, name, start_date, end_date, status, comments) Values ("PRJ003", "CLI003", "Website Design", "2019-07-02", "2019...   1 row(s) affected
120 00:27:12 Insert Into Project (proj_id, cl_id, name, start_date, status, comments) Values ("PRJ004", "CLI002", "Door Safety System", "2019-08-02", "on-progress...   1 row(s) affected
121 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ001", "EMP001", "2019-03-10 11:45:03", "Completed D...   1 row(s) affected
122 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ002", "EMP003", "2019-05-02 08:34:53", "Project starte...   1 row(s) affected
123 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ001", "EMP002", "2019-05-22 18:07:37", "Database Dr...   1 row(s) affected
124 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ002", "EMP003", "2019-06-01 08:12:34", "Microcontroll...   1 row(s) affected
125 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ003", "EMP001", "2019-07-02 12:30:22", "Project starte...   1 row(s) affected
126 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ002", "EMP004", "2019-08-01 08:00:00", "Project finalis...   1 row(s) affected
127 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ004", "EMP003", "2019-08-02 08:32:12", "Project starte...   1 row(s) affected
128 00:27:12 Insert Into ProjectProgress (proj_id, emp_id, daytime, action, completion, details) Values ("PRJ003", "EMP001", "2019-08-10 12:30:00", "Project compl...   1 row(s) affected
129 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001", "PRJ001", "Project Manager", "Responsible for the progress and completio...   1 row(s) affected
130 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP002", "PRJ001", "Developer", "Develop the database system")                             1 row(s) affected
131 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP003", "PRJ002", "Project Manager", "Responsible for the progress and completio...   1 row(s) affected
132 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP004", "PRJ002", "Developer", "Microcontroller Designer")                                1 row(s) affected
133 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001", "PRJ002", "Tester", "Responsible for the product being up to par")                1 row(s) affected
134 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP001", "PRJ003", "Sole Developer", "Responsible for the project")                        1 row(s) affected
135 00:27:12 Insert Into WorksOn (emp_id, proj_id, role, comments) Values ("EMP003", "PRJ004", "Sole Developer", "Responsible for the project")                        1 row(s) affected
136 00:27:12 Insert Into ProjectSize (proj_id, size, importance, no_dev) Values ("PRJ001", "M", "medium", "2")                                                          1 row(s) affected
137 00:27:12 Insert Into ProjectSize (proj_id, size, importance, no_dev, comments) Values ("PRJ002", "L", "medium", "3", "Finish in Three Months")                     1 row(s) affected
138 00:27:12 Insert Into ProjectSize (proj_id, size, importance, no_dev, comments) Values ("PRJ003", "S", "medium", "1", "Government query")                          1 row(s) affected
139 00:27:12 Insert Into ProjectSize (proj_id, size, importance, no_dev) Values ("PRJ004", "S", "low", "1")                                                             1 row(s) affected
140 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP001", "2019-04-30", "62.5", "29")                                             1 row(s) affected
141 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) Values ("EMP001", "2019-05-30", "62.5", "60", "10")                               1 row(s) affected
142 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-05-30", "35.2", "12")                                             1 row(s) affected
143 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-05-30", "35", "35")                                               1 row(s) affected
144 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) Values ("EMP001", "2019-06-30", "62.5", "60", "10")                               1 row(s) affected
145 00:27:12 Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-06-30", "35.2", "50")                                             1 row(s) affected
```



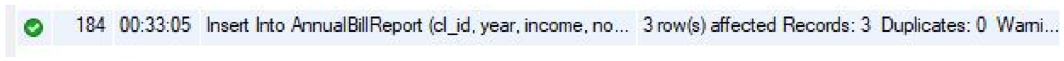
```
146  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-06-30", "35", "30")                          1 row(s) affected
147  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) Values ("EMP001", "2019-07-30", "62.5", "60", "20")             1 row(s) affected
148  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-07-30", "35.2", "45")                        1 row(s) affected
149  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-07-30", "35", "35")                          1 row(s) affected
150  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) Values ("EMP001", "2019-08-30", "62.5", "58", "5")              1 row(s) affected
151  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-08-30", "35.2", "50")                        1 row(s) affected
152  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-08-30", "35", "35")                          1 row(s) affected
153  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2019-08-30", "40", "5")                           1 row(s) affected
154  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours, bonus) Values ("EMP001", "2019-09-30", "62.5", "60", "10")             1 row(s) affected
155  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-09-30", "35.2", "45")                        1 row(s) affected
156  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-09-30", "35", "20")                          1 row(s) affected
157  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2019-09-30", "40", "25")                          1 row(s) affected
158  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP001", "2019-10-30", "62.5", "62")                        1 row(s) affected
159  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-10-30", "35.2", "55")                        1 row(s) affected
160  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-10-30", "35", "20")                          1 row(s) affected
161  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2019-10-30", "40", "15")                          1 row(s) affected
162  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP001", "2019-11-30", "62.5", "62")                        1 row(s) affected
163  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-11-30", "35.2", "60")                        1 row(s) affected
164  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-11-30", "35", "20")                          1 row(s) affected
165  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2019-11-30", "40", "15")                          1 row(s) affected
166  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP001", "2019-12-30", "62.5", "62")                        1 row(s) affected
167  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2019-12-30", "35.2", "60")                        1 row(s) affected
168  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2019-12-30", "35", "30")                          1 row(s) affected
169  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2019-12-30", "40", "15")                          1 row(s) affected
170  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP001", "2020-01-30", "65", "55")                          1 row(s) affected
171  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP002", "2020-01-30", "40", "50")                          1 row(s) affected
172  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP003", "2020-01-30", "35", "25")                          1 row(s) affected
173  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP004", "2020-01-30", "40", "15")                          1 row(s) affected
174  00:27:12  Insert Into Salary (emp_id, date_of_record, base_salary, hours) Values ("EMP005", "2020-01-30", "35", "20")                          1 row(s) affected
175  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL001", "PRJ001", "2019-03-10", "Database Design", "600", "100",...  1 row(s) affected
176  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL002", "PRJ002", "2019-05-10", "Design Completion", "350", "120...  1 row(s) affected
177  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL003", "PRJ003", "2019-05-02", "Deposit", "10", "50", "completed")  1 row(s) affected
178  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL004", "PRJ003", "2019-06-02", "Project Full Cost", "50", "140", "i...  1 row(s) affected
179  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL005", "PRJ001", "2019-07-02", "Deposit", "10", "50", "completed")  1 row(s) affected
180  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL006", "PRJ001", "2019-08-10", "Project Full Cost", "250", "50", "i...  1 row(s) affected
181  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL007", "PRJ004", "2019-08-03", "Deposit", "10", "50", "ready to is...  1 row(s) affected
182  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL008", "PRJ001", "2019-09-05", "Miscellaneous", "10", "50", "read...  1 row(s) affected
183  00:27:12  Insert Into Bill (bill_id, proj_id, daytime, work, hours, hourly_rate, status) Values ("BIL009", "PRJ003", "2020-01-04", "Maintenance", "5", "50", "ready t...  1 row(s) affected
```

All tables, except the AnnualBillReport Table, have been inserted with data. AnnualBillData is a special case. Because it is not and should not be allowed to insert into, it is auto-generated from the database yearly. The query below is the auto-generated query.

Auto-generated query:

```
-- Auto-generate AnnualBillReport using info from Client and Bill
Insert Into AnnualBillReport (cl_id, year, income, no_proj)
Select c.cl_id, year(daytime) as year, sum(total),
count(Distinct(b.proj_id))
From Client c Join Bill b Join Project p
Where c.cl_id = p.cl_id and b.proj_id = p.proj_id and b.status =
"completed"
Group by cl_id, year
Order by year asc;
```

Query ran:



```
184  00:33:05  Insert Into AnnualBillReport (cl_id, year, income, no...  3 row(s) affected Records: 3  Duplicates: 0  Warni...
```

IX. Views

Views are recorded queries that are stored in the database's dictionary to be executed right away when needed. Views do not store much data, but can almost be treated like a table (Create View in SQL Server 2020?). The use of views improves data management inside the database. The views used in Softtech's database is as mentioned below:

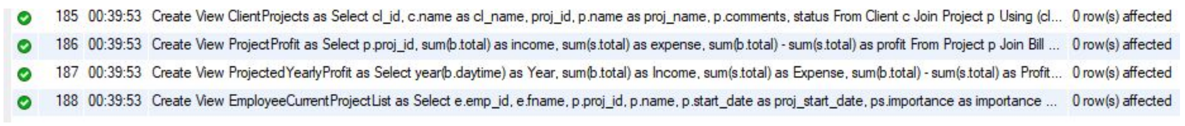
Views Queries:

```
-- This view shows a brief detail of the projects each client has.
-- It is intended to use for analytical purposes.
-- Management can use this to determine if a client is worth investing
more time into
Create View ClientProjects as
Select cl_id, c.name as cl_name, proj_id, p.name as proj_name,
p.comments, status
From Client c Join Project p Using (cl_id)
Order by c.cl_id;

-- This view shows how much profit each project has brought.
-- It can be used for analytical or confirmational purposes.
Create View ProjectProfit as
Select p.proj_id, sum(b.total) as income, sum(s.total) as expense,
sum(b.total) - sum(s.total) as profit
From Project p
Join Bill b Using (proj_id)
Join WorksOn w Using (proj_id)
Join Salary s On s.emp_id = w.emp_id
Where Date(s.date_of_record) Between p.start_date And p.end_date
And b.status = "completed"
Group by b.proj_id
Order by proj_id;

-- This view shows the predicted yearly profit from all the project
accepted
-- It will include all the project's bill regardless of it being
completed or not.
-- It can be used for analytical and planning purposes.
Create View ProjectedYearlyProfit as
Select year(b.daytime) as Year, sum(b.total) as Income, sum(s.total) as
Expense, sum(b.total) - sum(s.total) as Profit
From WorksOn w
```

```sql
Join Bill b Using (proj_id)
Join Salary s Using (emp_id)
Group by year(b.daytime)
Order by Year;

-- This view shows a list of project each employee has at the moment
-- Projects that have been completed will not be taken into account.
-- This can be used by managers to assign developers into new projects or
evaluate developer's workload
Create View EmployeeCurrentProjectList as
Select e.emp_id, e.fname, p.proj_id, p.name, p.start_date as
proj_start_date, ps.importance as importance
From WorksOn w
Join Employee e Using (emp_id)
Join Project p Using (proj_id)
Join ProjectSize ps Using (proj_id)
Where p.status != "completed"
Group by e.emp_id, p.proj_id
Order by e.emp_id asc;
```

The views can be put into great use for managers or analytics.

Queries ran:

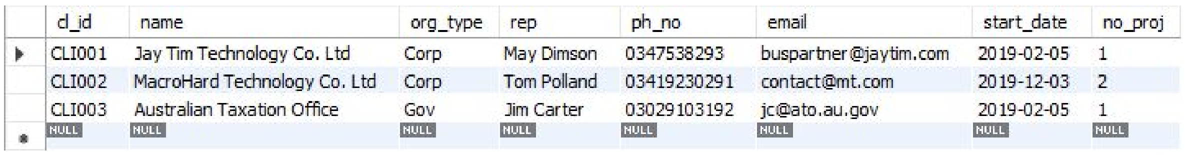


```
185  00:39:53  Create View ClientProjects as Select cl_id, c.name as cl_name, proj_id, p.name as proj_name, p.comments, status From Client c Join Project p Using (cl...  0 row(s) affected
186  00:39:53  Create View ProjectProfit as Select p.proj_id, sum(b.total) as income, sum(s.total) as expense, sum(b.total) - sum(s.total) as profit From Project p Join Bill ...  0 row(s) affected
187  00:39:53  Create View ProjectedYearlyProfit as Select year(b.daytime) as Year, sum(b.total) as Income, sum(s.total) as Expense, sum(b.total) - sum(s.total) as Profit...  0 row(s) affected
188  00:39:53  Create View EmployeeCurrentProjectList as Select e.emp_id, e.fname, p.proj_id, p.name, p.start_date as proj_start_date, ps.importance as importance ...  0 row(s) affected
```

X. Results

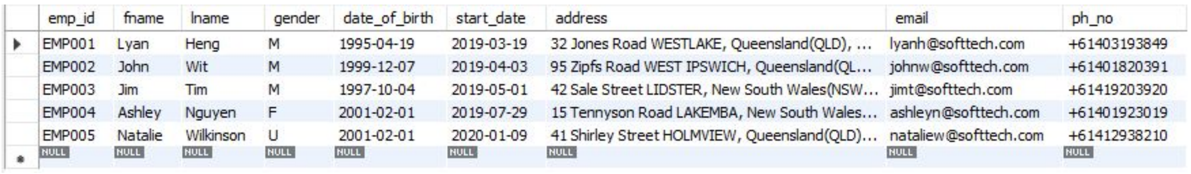
Results created:

Client:



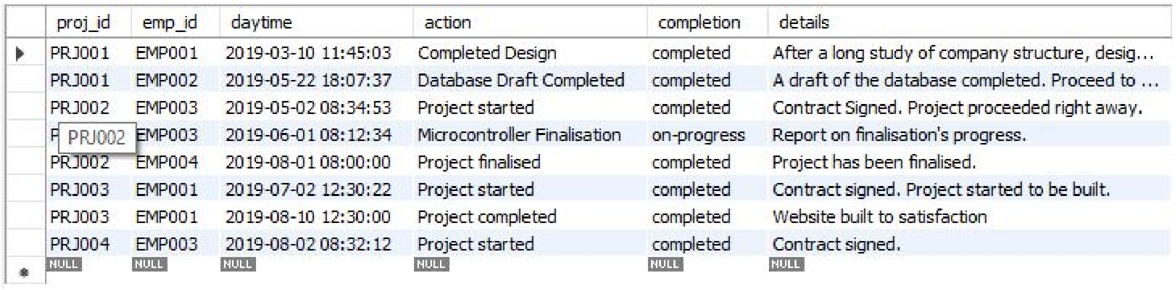
cl_id	name	org_type	rep	ph_no	email	start_date	no_proj
CLI001	Jay Tim Technology Co. Ltd	Corp	May Dimson	0347538293	buspartner@jaytim.com	2019-02-05	1
CLI002	MacroHard Technology Co. Ltd	Corp	Tom Polland	03419230291	contact@mt.com	2019-12-03	2
CLI003	Australian Taxation Office	Gov	Jim Carter	03029103192	jc@ato.au.gov	2019-02-05	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Employee:



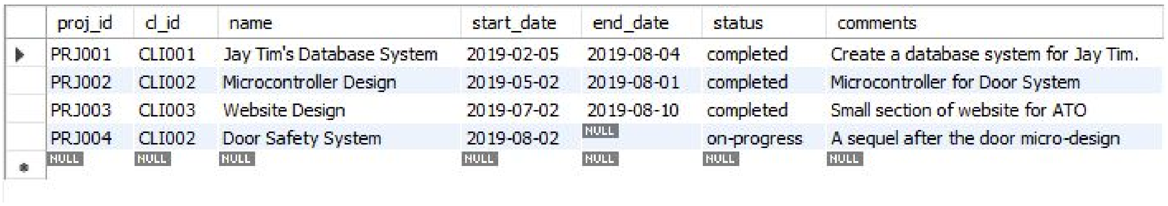
	emp_id	fname	lname	gender	date_of_birth	start_date	address	email	ph_no
▶	EMP001	Lyan	Heng	M	1995-04-19	2019-03-19	32 Jones Road WESTLAKE, Queensland(QLD), ...	lyanh@softtech.com	+61403193849
	EMP002	John	Wit	M	1999-12-07	2019-04-03	95 Zipfs Road WEST IPSWICH, Queensland(QL...	johnw@softtech.com	+61401820391
	EMP003	Jim	Tim	M	1997-10-04	2019-05-01	42 Sale Street LIDSTER, New South Wales(NSW...	jimt@softtech.com	+61419203920
	EMP004	Ashley	Nguyen	F	2001-02-01	2019-07-29	15 Tennyson Road LAKEMBA, New South Wales...	ashleyn@softtech.com	+61401923019
	EMP005	Natalie	Wilkinson	U	2001-02-01	2020-01-09	41 Shirley Street HOLMVIEW, Queensland(QLD)...	nataliew@softtech.com	+61412938210
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

ProjectProgress:



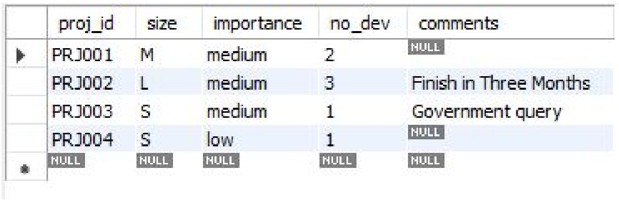
	proj_id	emp_id	daytime	action	completion	details
▶	PRJ001	EMP001	2019-03-10 11:45:03	Completed Design	completed	After a long study of company structure, desig...
	PRJ001	EMP002	2019-05-22 18:07:37	Database Draft Completed	completed	A draft of the database completed. Proceed to ...
	PRJ002	EMP003	2019-05-02 08:34:53	Project started	completed	Contract Signed. Project proceeded right away.
	PRJ002	EMP003	2019-06-01 08:12:34	Microcontroller Finalisation	on-progress	Report on finalisation's progress.
	PRJ002	EMP004	2019-08-01 08:00:00	Project finalised	completed	Project has been finalised.
	PRJ003	EMP001	2019-07-02 12:30:22	Project started	completed	Contract signed. Project started to be built.
	PRJ003	EMP001	2019-08-10 12:30:00	Project completed	completed	Website built to satisfaction
	PRJ004	EMP003	2019-08-02 08:32:12	Project started	completed	Contract signed.
*	NULL	NULL	NULL	NULL	NULL	NULL

Project:



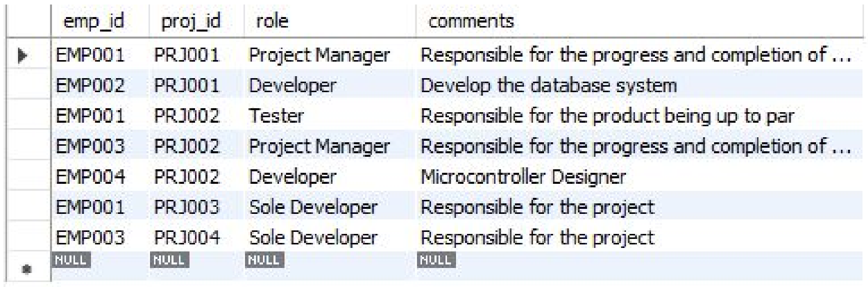
	proj_id	cl_id	name	start_date	end_date	status	comments
▶	PRJ001	CLI001	Jay Tim's Database System	2019-02-05	2019-08-04	completed	Create a database system for Jay Tim.
	PRJ002	CLI002	Microcontroller Design	2019-05-02	2019-08-01	completed	Microcontroller for Door System
	PRJ003	CLI003	Website Design	2019-07-02	2019-08-10	completed	Small section of website for ATO
	PRJ004	CLI002	Door Safety System	2019-08-02	NULL	on-progress	A sequel after the door micro-design
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

ProjectSize:



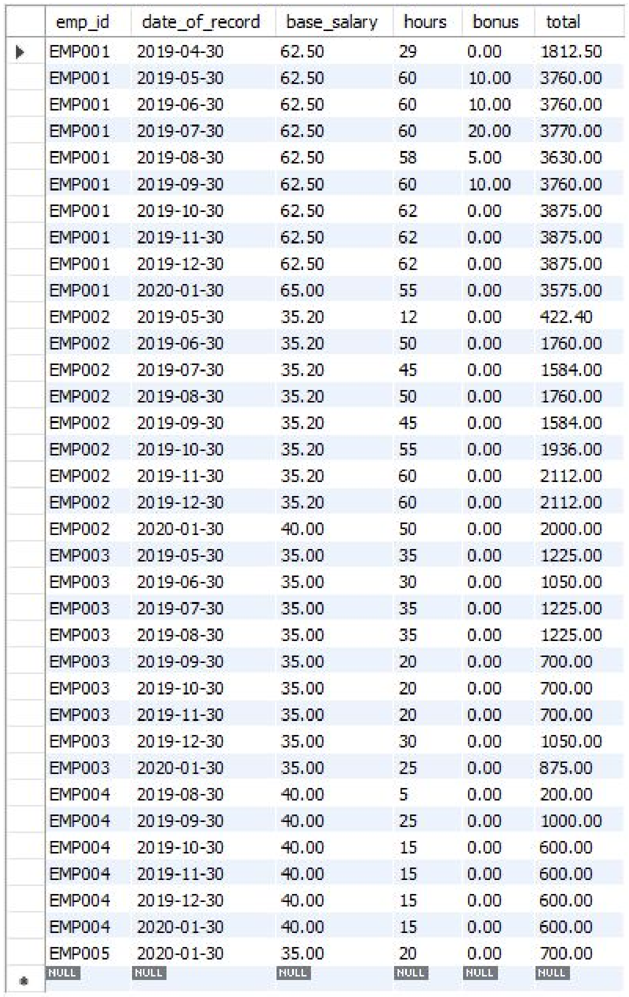
	proj_id	size	importance	no_dev	comments
▶	PRJ001	M	medium	2	NULL
	PRJ002	L	medium	3	Finish in Three Months
	PRJ003	S	medium	1	Government query
	PRJ004	S	low	1	NULL
*	NULL	NULL	NULL	NULL	NULL

WorksOn:



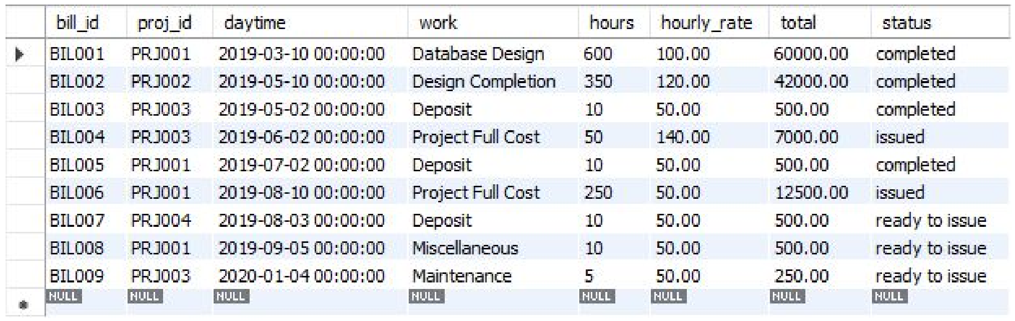
	emp_id	proj_id	role	comments
▶	EMP001	PRJ001	Project Manager	Responsible for the progress and completion of ...
	EMP002	PRJ001	Developer	Develop the database system
	EMP001	PRJ002	Tester	Responsible for the product being up to par
	EMP003	PRJ002	Project Manager	Responsible for the progress and completion of ...
	EMP004	PRJ002	Developer	Microcontroller Designer
	EMP001	PRJ003	Sole Developer	Responsible for the project
	EMP003	PRJ004	Sole Developer	Responsible for the project
*	NULL	NULL	NULL	NULL

Salary:



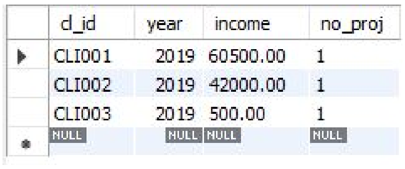
	emp_id	date_of_record	base_salary	hours	bonus	total
▶	EMP001	2019-04-30	62.50	29	0.00	1812.50
	EMP001	2019-05-30	62.50	60	10.00	3760.00
	EMP001	2019-06-30	62.50	60	10.00	3760.00
	EMP001	2019-07-30	62.50	60	20.00	3770.00
	EMP001	2019-08-30	62.50	58	5.00	3630.00
	EMP001	2019-09-30	62.50	60	10.00	3760.00
	EMP001	2019-10-30	62.50	62	0.00	3875.00
	EMP001	2019-11-30	62.50	62	0.00	3875.00
	EMP001	2019-12-30	62.50	62	0.00	3875.00
	EMP001	2020-01-30	65.00	55	0.00	3575.00
	EMP002	2019-05-30	35.20	12	0.00	422.40
	EMP002	2019-06-30	35.20	50	0.00	1760.00
	EMP002	2019-07-30	35.20	45	0.00	1584.00
	EMP002	2019-08-30	35.20	50	0.00	1760.00
	EMP002	2019-09-30	35.20	45	0.00	1584.00
	EMP002	2019-10-30	35.20	55	0.00	1936.00
	EMP002	2019-11-30	35.20	60	0.00	2112.00
	EMP002	2019-12-30	35.20	60	0.00	2112.00
	EMP002	2020-01-30	40.00	50	0.00	2000.00
	EMP003	2019-05-30	35.00	35	0.00	1225.00
	EMP003	2019-06-30	35.00	30	0.00	1050.00
	EMP003	2019-07-30	35.00	35	0.00	1225.00
	EMP003	2019-08-30	35.00	35	0.00	1225.00
	EMP003	2019-09-30	35.00	20	0.00	700.00
	EMP003	2019-10-30	35.00	20	0.00	700.00
	EMP003	2019-11-30	35.00	20	0.00	700.00
	EMP003	2019-12-30	35.00	30	0.00	1050.00
	EMP003	2020-01-30	35.00	25	0.00	875.00
	EMP004	2019-08-30	40.00	5	0.00	200.00
	EMP004	2019-09-30	40.00	25	0.00	1000.00
	EMP004	2019-10-30	40.00	15	0.00	600.00
	EMP004	2019-11-30	40.00	15	0.00	600.00
	EMP004	2019-12-30	40.00	15	0.00	600.00
	EMP004	2020-01-30	40.00	15	0.00	600.00
	EMP005	2020-01-30	35.00	20	0.00	700.00
*	NULL	NULL	NULL	NULL	NULL	NULL

Bill:



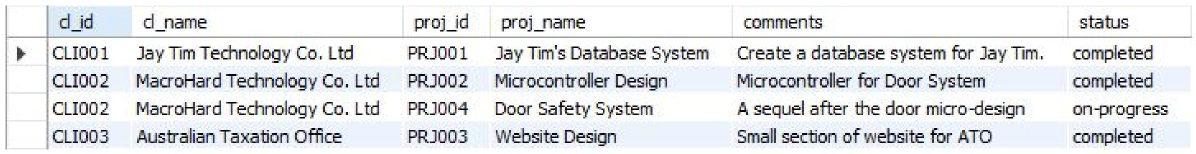
bill_id	proj_id	daytime	work	hours	hourly_rate	total	status
BIL001	PRJ001	2019-03-10 00:00:00	Database Design	600	100.00	60000.00	completed
BIL002	PRJ002	2019-05-10 00:00:00	Design Completion	350	120.00	42000.00	completed
BIL003	PRJ003	2019-05-02 00:00:00	Deposit	10	50.00	500.00	completed
BIL004	PRJ003	2019-06-02 00:00:00	Project Full Cost	50	140.00	7000.00	issued
BIL005	PRJ001	2019-07-02 00:00:00	Deposit	10	50.00	500.00	completed
BIL006	PRJ001	2019-08-10 00:00:00	Project Full Cost	250	50.00	12500.00	issued
BIL007	PRJ004	2019-08-03 00:00:00	Deposit	10	50.00	500.00	ready to issue
BIL008	PRJ001	2019-09-05 00:00:00	Miscellaneous	10	50.00	500.00	ready to issue
BIL009	PRJ003	2020-01-04 00:00:00	Maintenance	5	50.00	250.00	ready to issue
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

AnnualBillReport:



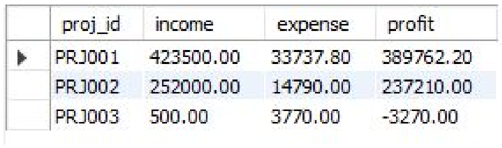
cl_id	year	income	no_proj
CLI001	2019	60500.00	1
CLI002	2019	42000.00	1
CLI003	2019	500.00	1
NULL	NULL	NULL	NULL

ClientProject (View):



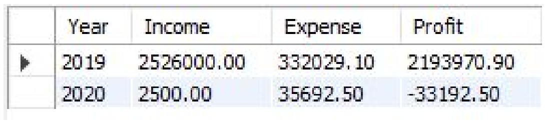
cl_id	cl_name	proj_id	proj_name	comments	status
CLI001	Jay Tim Technology Co. Ltd	PRJ001	Jay Tim's Database System	Create a database system for Jay Tim.	completed
CLI002	MacroHard Technology Co. Ltd	PRJ002	Microcontroller Design	Microcontroller for Door System	completed
CLI002	MacroHard Technology Co. Ltd	PRJ004	Door Safety System	A sequel after the door micro-design	on-progress
CLI003	Australian Taxation Office	PRJ003	Website Design	Small section of website for ATO	completed

ProjectProfit (View):



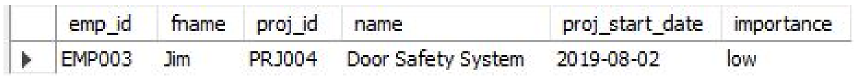
proj_id	income	expense	profit
PRJ001	423500.00	33737.80	389762.20
PRJ002	252000.00	14790.00	237210.00
PRJ003	500.00	3770.00	-3270.00

ProjectedYearlyProfit (View):



Year	Income	Expense	Profit
2019	2526000.00	332029.10	2193970.90
2020	2500.00	35692.50	-33192.50

EmployeeCurrentProjectList (View):



emp_id	fname	proj_id	name	proj_start_date	importance
EMP003	Jim	PRJ004	Door Safety System	2019-08-02	low

XI. Conclusion

The database creation project is a success. The structure of the company was closely studied, database design carefully designed, and database effectively implemented and constructed. The result shows the success of all the queries used to build the database. The flexibility of the database is also enhanced mainly with the design and implementation of auto-generation and triggers.

XII. References

- Brumm, B 2017, "How to Handle a Many-to-Many Relationship in Database Design", Database Zone, viewed 29 October, 2020, <https://dzone.com/articles/how-to-handle-a-many-to-many-relationship-in-datab>.

- Chapple, M 2020, "Database Normalization Basics", LifeWire, viewed 29 October, 2020, <https://www.lifewire.com/database-normalization-basics-1019735>.

- Create View in SQL Server 2020?, Tutorial Gateway, viewed 30 October 2020, <https://www.tutorialgateway.org/views-in-sql-server/>.