University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___1_____

Date: _____2/7/2023_____

Group Number: __195_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Enoch Cheung	21988233	o4n2b	ajienoch@gmail.com
Ryan Clayton	96445416	k7p4n	ryanclayton31@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

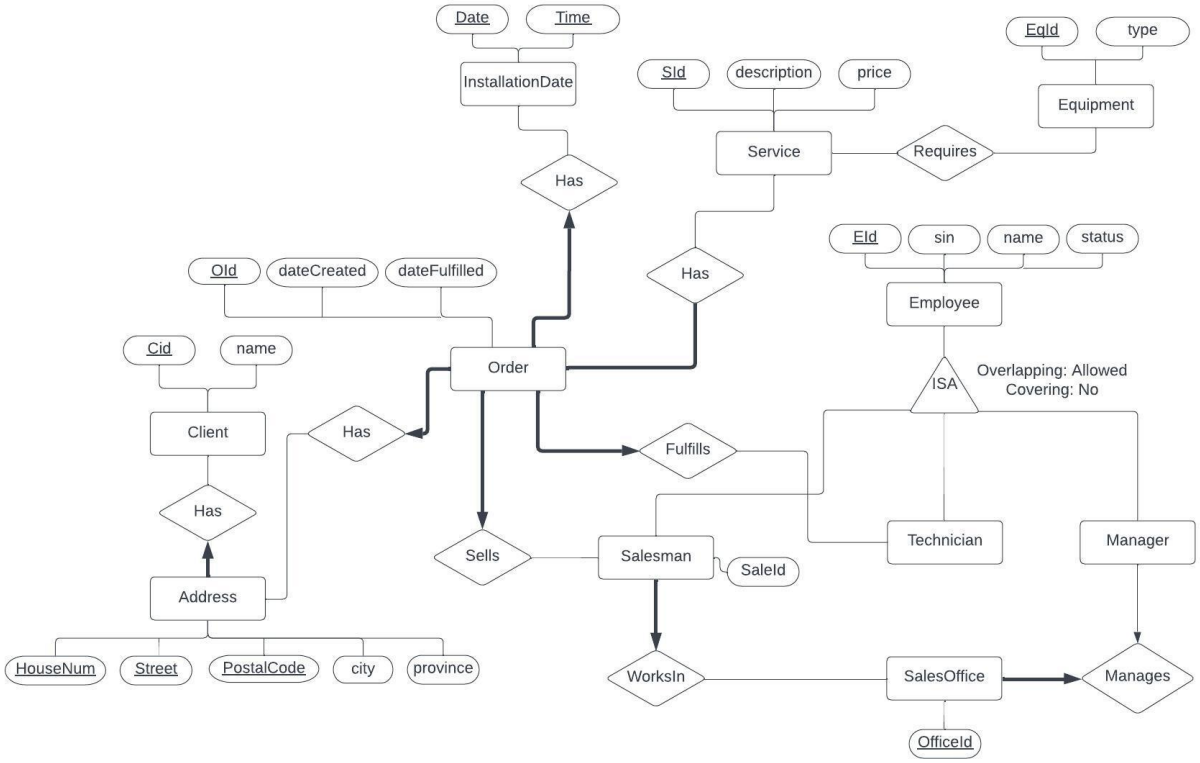
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Milestone 2

2. A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects so this will help them remember details about your project.

Our application resides within the door-2-door sales industry for alarm & solar services. Door-2-door sales refers to sales representatives meeting homeowners and selling products at their doorstep. We aim to create a system to track the sales process for stakeholders such as salesmen, managers, and technicians.

3. The ER diagram you are basing your item #3 (below) on. This ER diagram may be the same as your milestone 1 submission or it might be different. If you have made changes from the version submitted in milestone 1, attach a note indicating what changes have been made and why.



4. The schema derived from your ER diagram (above). For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:

a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)).
b. Make sure to include the domains for each attribute.
c. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain

Table Definitions:

Tables	**Table definition**
Client	Client(*client_id*: int(12) (PK), *name*: CHAR[30] NOT NULL)
Address	Address(*HouseNum*: int(12) (PK), *Street*: CHAR[30] (PK), *postalCode*: *CHAR*(6) (PK), *City:* CHAR[30] NOT NULL, *province*: CHAR[2] NOT NULL, *Cid*: int(12) NOT NULL (FK))
Order	Order(*Oid*: int(12) (PK), *dateCreated* : DATETIME NOT NULL, *dateFulfilled* : DATETIME, *Date* : DATE NOT NULL (FK), *Time* : DATETIME NOT NULL (FK), *HouseNum* : int(12) NOT NULL (FK),

	Street: CHAR[30] NOT NULL (FK), PostalCode: CHAR[6] NOT NULL (FK) SaleId: int(12) NOT NULL (FK), SId: int(32) NOT NULL (FK), EId: int(32) (FK))
Employee	Employee(*EId*: int(6) (PK), *sin* : int(9) UNIQUE NOT NULL (CK), *name*: CHAR[30] NOT NULL, *status*: CHAR[1] NOT NULL)
Technician	Technician(*Eid*: int(6) (PK, FK))
Manager	Manager(*Eid*: int(6) (PK, FK))
Salesman	Salesman(*Eid*: int(6) (PK, FK), *SaleId*: int(12) UNIQUE NOT NULL (CK), *OfficeId*: int(6) NOT NULL (FK))
Equipment	Equipment(*EqId*: int(12) (PK), *type*: CHAR(20))
Service	Service(*Sid*: int(12) (PK), *description*: CHAR[30] NOT NULL, *price*: int(6) NOT NULL, *EqId*: int(12) NOT NULL (FK))

Installation Date	InstallationDate(*Date*: DATE (PK), *Time*: DATETIME (PK))
SalesOffice	SalesOffice(*OfficeId*: int(6) (PK), *Eid*: int(6) UNIQUE NOT NULL (FK))

5. Functional Dependencies (FDs)

a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key). PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A → A. Note: In your list of FDs, there must be some kind of valid FD other than those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process.

Client:

Cid -> name

Address:

HouseNum, Street, PostalCode -> city, province
PostalCode -> city, province

Order:

Oid -> (*dateCreated, dateFulfilled, totalPrice*, *Date, Time*, *HouseNum, Street*, PostalCode, SaleID)

Employee:

sin -> Eid, name, status
Eid -> sin, name, status

Technician:

No need for a standalone table as the employees table is referenced.

Manager:

No need for a standalone table since SalesOffice references Employees table through Employees (Eid).

Salesman:

SaleId -> Eid
Eid -> SaleId

Equipment:

EqId -> type

Service:

Sid -> description, price, EqId

Installation Date:

N/A

SalesOffice:

OfficeId -> Eid
Eid -> OfficeId

6. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization. You should show the steps taken for the decomposition. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, …). ALL Tables must be listed, not only the ones post normalization.

Normalize address:

HouseNum, Street, PostalCode -> city, province
PostalCode -> city, province

Tables	Table definition
Address	Address(*HouseNum*: int(12) (PK), *Street*: CHAR[30] (PK), *postalCode*: *CHAR*(6) (PK, FK), Cid: int(12) NOT NULL (FK))
PostalCode	PostalCode(*postalCode*: *CHAR*(6) (PK), *City:* CHAR[30] NOT NULL, *province*: CHAR[2] NOT NULL,)

All other tables are as follows from 4.

7. The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc.

Client:

```
CREATE TABLE Client(
        client_id: int(12),
        name: CHAR(30) NOT NULL,
        PRIMARY KEY(client_id)
)
```

Address:

```
Create Table Address (

        HouseNum: int(12),
        Street: CHAR(30) (PK),
        postalCode: CHAR(6) ,
        Cid: int(12) NOT NULL,
        PRIMARY KEY (HouseNum, Street, postalCode),
        FOREIGN KEY (postalCode) references PostalCode,
        FOREIGN KEY (Cid) references Client
```

)

PostalCode (
    postalCode: CHAR(6),
    City: CHAR(30) NOT NULL,
    province: CHAR(2) NOT NULL,
    PRIMARY KEY (postalCode)
)

Order:
CREATE TABLE Order (
    Oid: int(12) (PK),
    dateCreated : DATETIME NOT NULL,
    dateFulfilled : DATETIME,
    Date : DATE NOT NULL,
    Time : DATETIME NOT NULL,
    HouseNum : int(12) NOT NULL,
    Street: CHAR(30) NOT NULL,
    postalCode: CHAR(6) NOT NULL,
    SaleId: int(12) NOT NULL,
    SId: int(12) NOT NULL,
    EId: int(12),
    PRIMARY KEY (Oid),
    FOREIGN KEY (Date, Time) references InstallationDate,
    FOREIGN KEY (HouseNum, Street, postalCode) references Address,
    FOREIGN KEY (SaleId) references Salesman,
    FOREIGN KEY (SId) references Service
    FOREIGN KEY (EId) references Employee
)

Employee:
CREATE TABLE Employee(
    EId: int(6),
    sin : int(9) UNIQUE NOT NULL,
    name: CHAR(30) NOT NULL,
    status: CHAR(1) NOT NULL,
    PRIMARY KEY (EId)
)

Technician:

No need for a standalone table as the employees table is referenced.

Manager:

No need for a standalone table since SalesOffice references Employees table through Employees (Eid).

Salesman:

CREATE TABLE Salesman(

Eid: int(6) (PK, FK),
SaleId: int(12) UNIQUE NOT NULL,
OfficeId: int(6) NOT NULL
PRIMARY KEY (*Eid*) REFERENCES Employees
FOREIGN KEY(*OfficeIdI*) REFERENCES SalesOffice

)

Equipment:

CREATE TABLE Equipment(

EqId: int(12),
type: CHAR(20),
PRIMARY KEY (EqId)

)

Service:

CREATE TABLE Service (

Sid: int(12) (PK),
description: CHAR[30] NOT NULL,
price: int(6) NOT NULL,
EqId: int(12) NOT NULL
PRIMARY KEY (Sid)
FOREIGN KEY (EqId) REFERENCES Employees

)

Installation Date:

CREATE TABLE InstallationDate (

Date: DATE,
Time: DATETIME,
PRIMARY KEY (Date, Time)

)

SalesOffice:
CREATE TABLE SalesOffice (
    OfficeId: int(6),
    Eid: int(6) UNIQUE NOT NULL,
    PRIMARY KEY (OfficeId),
    FOREIGN KEY (Eid) REFERENCES Employees
)

8. INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later on.

Client
INSERT INTO Client (client_id, name)
VALUES('001', 'Rob')

INSERT INTO Client (client_id, name)
VALUES('002', 'Gene')

INSERT INTO Client (client_id, name)
VALUES('003', 'Samantha')

INSERT INTO Client (client_id, name)
VALUES('004', 'Max')

INSERT INTO Client (client_id, name)
VALUES('005', 'Tyler')

Address
INSERT INTO Address (HouseNum, Street, postalCode, Cid)
VALUES('123456', 'Baker', 'V9P 0A5', '001')

INSERT INTO Address (HouseNum, Street, postalCode, Cid)
VALUES('123457', 'Baker', 'V9P 0A5', '002')

INSERT INTO Address (HouseNum, Street, postalCode, Cid)
VALUES('123458', 'Baker', 'V9P 0A5', '003')

INSERT INTO Address (HouseNum, Street, postalCode, Cid)
VALUES(‘123459’, ‘Baker’, ‘V9P 0A5’, ‘004’)

INSERT INTO Address (HouseNum, Street, postalCode, Cid)
VALUES(‘12340’, ‘Baker’, ‘V9P 0A5’, ‘005’)

<u>PostalCode</u>
INSERT INTO PostalCode (postalCode, City, province)
VALUES(‘V9P 0A5’, ‘Parksville’, ‘BC)

INSERT INTO PostalCode (postalCode, City, province)
VALUES(‘V8J 5V1’, ‘Prince Rupert’, ‘BC’)

INSERT INTO PostalCode (postalCode, City, province)
VALUES(‘G6B 4B5’, ‘Lac-Megantic’, ‘QC’)

INSERT INTO PostalCode (postalCode, City, province)
VALUES(‘T7Z 8N3’, ‘Stony Plain’, ‘AB’)

INSERT INTO PostalCode (postalCode, City, province)
VALUES(‘E3E 5T2’, ‘Kingsclear’, ‘NB’)

<u>Order</u>
INSERT INTO Order (Oid, dateCreated, dateFulfilled, Date, Time, HouseNum, Street, postalCode, SaleId, SId, EId)
VALUES(‘000000000001’, ‘2023-01-01’, ‘2023-01-01 00:00:00’, ‘123456’, ‘Baker’, ‘V9P 0A5’, ‘000001’, ’1’, ‘000006’)

INSERT INTO Order (Oid, dateCreated, dateFulfilled, Date, Time, HouseNum, Street, postalCode, SaleId, SId, EId)
VALUES(‘000000000002’, ‘2023-01-01’, ‘2023-01-01 00:02:00’, ‘123457’, ‘Baker’, ‘V9P 0A5’, ‘000002’, ’2’, , ‘000006’)

INSERT INTO Order (Oid, dateCreated, dateFulfilled, Date, Time, HouseNum, Street, postalCode, SaleId, SId, EId)
VALUES(‘000000000003’, ‘2023-01-01’, ‘2023-01-01 00:02:00’, ‘123458’, ‘Baker’, ‘V9P 0A5’, ‘000003’, ’3’, , ‘000006’)

INSERT INTO Order (Oid, dateCreated, dateFulfilled, Date, Time, HouseNum, Street, postalCode, SaleId, SId, EId)
VALUES(‘000000000003’, ‘2023-01-01’, ‘2023-01-01 00:02:00’, ‘123459’, ‘Baker’, ‘V9P 0A5’, ‘000004’, ’4’, , ‘000006’)

INSERT INTO Order (Oid, dateCreated, dateFulfilled, Date, Time, HouseNum, Street, postalCode, SaleId, SId, EId)
VALUES(‘000000000003’, ‘2023-01-01’, ‘2023-01-01 00:02:00’, ‘123450’, ‘Baker’, ‘V9P 0A5’, ‘000005’, ’5’, , ‘000006’)

<u>Employee:</u>
CREATE TABLE Employee(
    EId: int(6),
    sin : int(9) UNIQUE NOT NULL,
    name: CHAR[30] NOT NULL,
    status: CHAR[1] NOT NULL,
    PRIMARY KEY (EId)
)
INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000001’, ‘100000001’, ‘Tyler’, ‘Y’)

INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000002’, ‘100000002’, ‘Sam’, ‘Y’)

INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000003’, ‘100000003’, ‘George’, ‘Y’)

INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000004’, ‘100000003’, ‘Michelle’, ‘Y’)

INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000005’, ‘100000005’, ‘Reagan’, ‘Y’)

INSERT INTO Employees (EId, sin, name, status)
VALUES (‘000006’, ‘100000006’, ‘Steve’, ‘T’)

<u>Salesmen:</u>
INSERT INTO Salesman (Eid, SaleId, OfficeId)
VALUES (‘000001’, ‘000001’,999999)

INSERT INTO Salesman (Eid, SaleId, OfficeId)
VALUES (‘000002’, ‘000002’,999998)

INSERT INTO Salesman (Eid, SaleId, OfficeId)
VALUES (‘000003’, ‘000003’,999997)

INSERT INTO Salesman (Eid, SaleId, OfficeId)
VALUES (‘000004’, ‘000004’,999996)

INSERT INTO Salesman (Eid, SaleId, OfficeId)
VALUES (‘000005’, ‘000005’,999995)

Equipment:
INSERT INTO Equipment (EqId, type)
VALUES (‘1’, ‘Toolset 1’)

INSERT INTO Equipment (EqId, type)
VALUES (‘2’, ‘Toolset 2’)

INSERT INTO Equipment (EqId, type)
VALUES (‘3’, ‘Toolset 3’)

INSERT INTO Equipment (EqId, type)
VALUES (‘4’, ‘Toolset 4’)

INSERT INTO Equipment (EqId, type)
VALUES (‘5’, ‘Toolset 5’)

Service:
INSERT INTO Service (Sid, description, price, EqId)
VALUES (‘1’, ‘Package 1’, ‘500’, ‘1’)

INSERT INTO Service (Sid, description, price, EqId)
VALUES (‘2’, ‘Package 2’, ‘1000’, ‘2’)

INSERT INTO Service (Sid, description, price, EqId)
VALUES (‘3’, ‘Package 3’, ‘1500’, ‘3’)

INSERT INTO Service (Sid, description, price, EqId)
VALUES (‘4’, ‘Package 4’, ‘2500’, ‘4’)

INSERT INTO Service (Sid, description, price, EqId)
VALUES (‘5’, ‘Package 5’, ‘3500’, ‘5’)

Installation Date:
INSERT INTO InstallationDate (Date, Time)
VALUES (‘2023-01-01’, ‘2023-01-01 12:00:00’)

INSERT INTO InstallationDate (Date, Time)
VALUES (‘2023-01-01’, ‘2023-01-01 15:00:00’)

INSERT INTO InstallationDate (Date, Time)
VALUES (‘2023-01-02’, ‘2023-01-02 9:00:00’)

INSERT INTO InstallationDate (Date, Time)
VALUES (‘2023-01-03’, ‘2023-01-03 13:00:00’)

INSERT INTO InstallationDate (Date, Time)
VALUES (‘2023-01-04’, ‘2023-01-04 18:30:00’)

SalesOffice:
INSERT INTO SalesOffice (OfficeId, Eid)
VALUES (‘999999’, ‘000001’);

INSERT INTO SalesOffice (OfficeId, Eid)
VALUES (‘999998’, ‘000002’);

INSERT INTO SalesOffice (OfficeId, Eid)
VALUES (‘999997’, ‘000003’);

INSERT INTO SalesOffice (OfficeId, Eid)
VALUES (‘999996’, ‘000004’);

INSERT INTO SalesOffice (OfficeId, Eid)
VALUES (‘999995’, ‘000005’);