

COMP 5531

Files and Databases

Summer 2018, Section BB, Group EBC5531_1

Warm Up Project

Report

June 21, 2018

<u>Student Name</u>	<u>Id</u>
Faezeh Mobasheri	26821022
Jasdeep Ratot	26383599
Benardo Sandi Morales	40046377
Aida Sharif Rohani	21341669
Stefan Wapnick	26211798

Contents

1	Submission Overview	3
2	Connection Information	3
3	ER Diagram	4
3.1	Justification and Alternatives Analysis	5
3.1.1	Selection of Primary Key	5
3.1.2	Representation of Enumeration Type Properties as Attributes	5
3.1.3	Employee Name Attribute	5
3.1.4	Representing Sales Associates and Managers by WorksOn Relationship	5
3.1.5	Manager Role	6
4	Tables and Data Types Design	7
4.1	Justification and Alternatives Analysis	8
4.1.1	Enumeration Data Type	8
4.1.2	Decimal Type to Represent Money	8
4.1.3	Phone Field Representation	8
5	Scripts	9
5.1	Creation Script	9
5.2	Insert Data Script	10
5.3	Queries	12
5.3.1	List of Employees Working Exclusively on Cloud Contracts	12
5.3.2	Employee Ids of all developers	12
5.3.3	Employees Working on Contract with ACV of at least \$20,000	12
5.3.4	Contracts Managed by Van David	13
6	Assumptions	13
7	Distribution of Work	14
	References	14
	Appendix A –Tables Data Printouts	15
	Appendix B – Query Results Printouts	18

1 Submission Overview

This submission contains the ER model, database tables, mock data and sample queries for a basic Contract Management System (CMS) application implemented in MySQL.

Table 1 details the main contents of the warm up project submission

Table 1 – Submission files

File or Folder	Description
create.sql	Script to create tables. This script will also delete any existing tables prior to running.
insert.sql	Script to insert test data into the database. This script will also clean any existing records prior to running.
queries.sql	Script containing warm-up project queries.
ERDiagram.pdf	Contains the ER diagram of the contract management system using Chen notation.
tables.pdf	Visualization of tables created from ER diagram.
tableData/	Folder containing exports of the data inserted into the database tables (using insert.sql script).
queryResults/	Folder containing results of queries from running queries.sql.

2 Connection Information

The scripts have been deployed and tested on the ENCS hosted database server (ebc5531.encs.concordia.ca). The following connection information was used for testing.

Table 2 – Database server connection information for testing

Field	Value
Server	ebc5531.encs.concordia.ca
Username	ebc55311
Database	ebc55311
Password	b1gB3ars
Sample MySQL shell connection	mysql -h ebc5531.encs.concordia.ca -u ebc55311 -p ebc55311

3 ER Diagram

Figure 1 shows the ER diagram for the Contract Management System.

COMP 5531

Contract Management System - ER Diagram

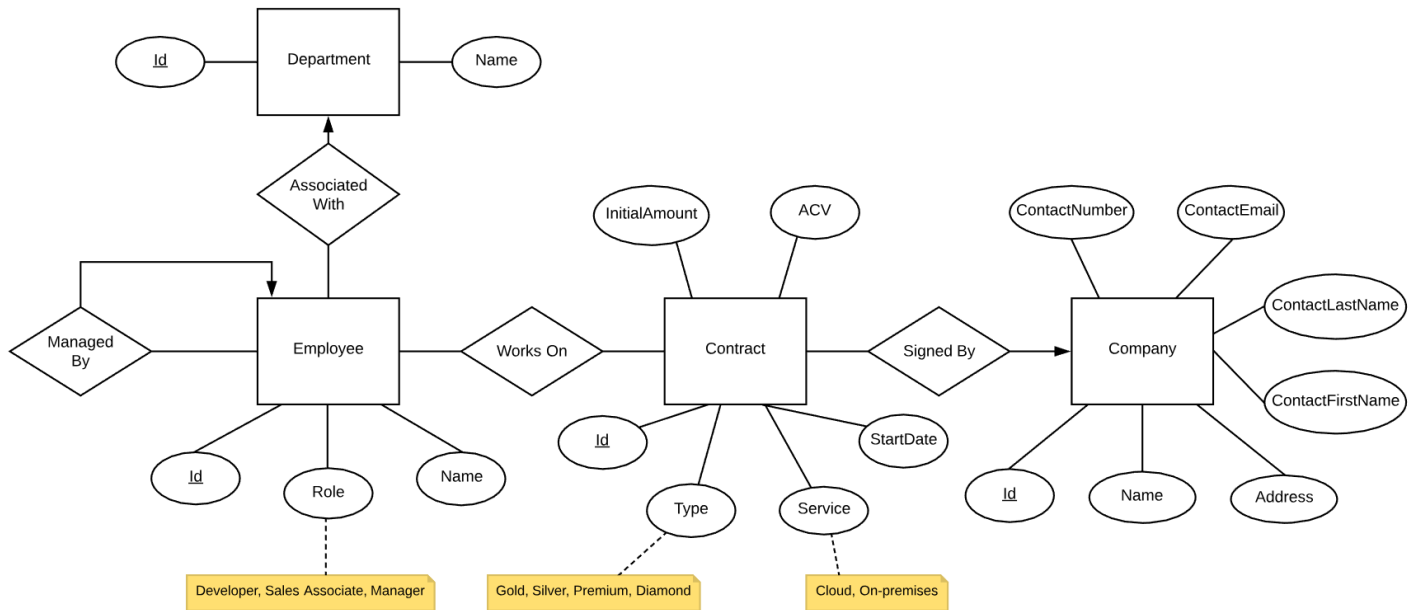


Figure 1 – ER Diagram for Contract Management System



By convention, the foreign keys are not shown as attributes in the ER diagram entities and are implied from the depicted relationships symbols (diamonds). They can be found in the database tables diagram in Figure 2.

The ER diagram represents the relationship between employees, departments, contracts and the companies that sign these constraints. The Works On relationship is used to represent all employees who work on a contract (Developer, Sales Associate and Manager). It is important to note that there are two layers of management: employees managed by a manager and management by a manager of contracts.

The relationship of employees grouped under managers in a department is represented by the “Managed By” relationship in the Employee entity. Conversely, the relationship of managers managing a project is represented by the Works On relationship between the Employee and Contract entity.

3.1 Justification and Alternatives Analysis

Some important alternatives encountered when designing the ER diagram are discussed here and in each case the justification for choosing the current model is given.

3.1.1 Selection of Primary Key

An additional integer, unique auto-increment Id field is defined to represent the primary key of records instead of using existing attributes to define the primary key. Defining a monotonically increasing integer as the primary key ensures uniqueness and natural ordering of records which is desired for the primary key as it determines the way records are stored physically. A small numeric id field may also help performance when doing joins as opposed to a heavier string-based field.

3.1.2 Representation of Enumeration Type Properties as Attributes

Enumeration type properties (such as Employee.Role, Contract.Type, Contract.Service) were chosen to be represented as attributes on their respective parent entities instead of placing them in their own entity. It could be argued that this introduces some data redundancy (for example, updating a role string would require updating all employee records with that role). To avoid this problem, these enumeration values could have been placed in their own tables and linked to their parent entities via foreign key constraint. Despite this, it was chosen to represent these properties with an attribute instead for the following reasons:

- Placing them in another entity would result in another table and would require an extra join to query by the string value of the enumeration, hurting performance and making the queries significantly more complicated to read.
- It is assumed that these values will not change and so they can essentially be viewed as like constant ids, in which case previous concerns regarding data redundancy in terms of updates or deletions becomes less relevant.

3.1.3 Employee Name Attribute

It was decided to use a single attribute to represent the employee name instead of using two attributes: first name and last name. This was done because all queries as of present treat the name field as one and so this data representation is more convenient for the current set of queries. Depending on future requirements this structure may be changed.

3.1.4 Representing Sales Associates and Managers by WorksOn Relationship

The current design represents all employee relationships with contracts using the many-to-many WorksOn relationship (including managers and sales associates). It could be argued that it might be more appropriate to instead use foreign keys on the Contract entity pointing to the Employee table for

managers and sales associates since there is conventionally only one manager and one sales associate per contract and thus the relationship for sales associate and manager is many-to-one and not many-to-many and as the Works On relationship indicates. In short, the many-to-many joining table is only required for Developer type employees.

However, it was decided to also represent Managers and Sales Associate type employees' relationship with contracts via the many-to-many Works On joining table to treat the data uniformly. For example, the query "get all employees working on a contract" can be more easily done with the single Works On relationship representing instead of having to consider multiple relationship associations and combine the query results of each. Employees of a specific role in a contract can still be easily determined with an additional where clause containing the role. Likewise, although the relationship of Managers and Sales Associates to contracts is not many-to-many, having the infrastructure in place to make it so in the future lends flexibility to the design.

3.1.5 Manager Role

Assuming the managers themselves have no manager, it could be argued that the Manager role is somewhat redundant since we can find who is a manager by looking at Employee.ManagerId column and checking if it is null. However, the Role attribute is required nonetheless for Developer and Sales Associates employees and so in this case a manager employee would have a null Role which is somewhat awkward. For this reason and for the sake of more clear queries, the Manager role was added nonetheless.

4 Tables and Data Types Design

Based on the ER diagram of Figure 1, all entities were translated to database tables with the addition of an extra WorksOn table to support the many-to-many relationship between Employee and Contract entities. The creation script itself can be found in the Scripts section.

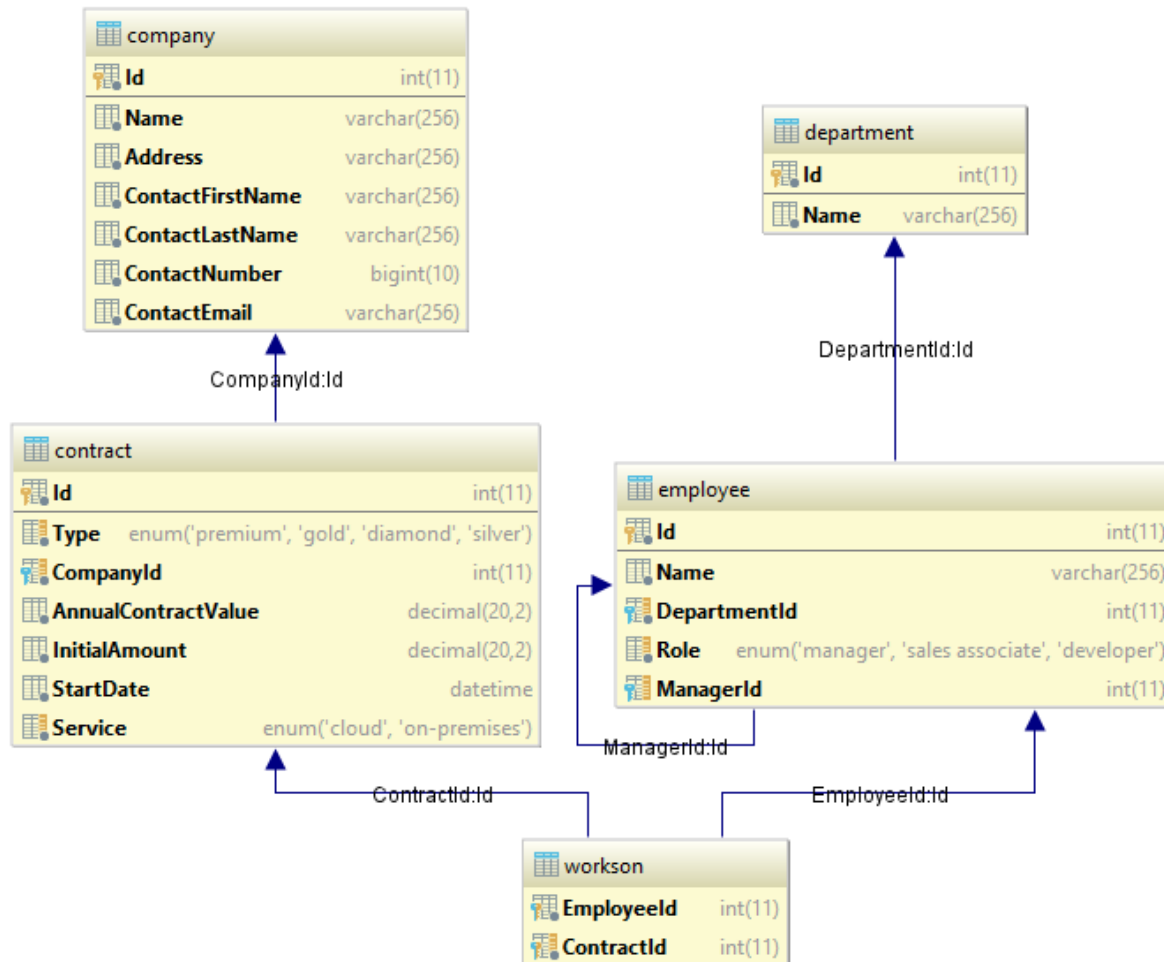


Figure 2 – Tables and data types (generated by IDE)



MySQL shows int(11) when describing int fields however these columns are created as simply INT in the create.sql script.

When translated to database tables the following foreign key constraints were created:

Table 3 - Foreign key relationships in tables

Constraint	Multiplicity	Foreign Keys (FK)
Employee to Manager	Many-to-One	Employee.ManagerId FK points to Employee.Id of manager
Employee to contract(s) worked on	Many-to-Many	WorksOn.EmployeeId FK points to Employee.Id WorksOn.ContractId FK points to Contract.Id
Contract to Company	Many-to-One	Contract.CompanyId FK points to Company.Id
Employee to Department	Many-to-One	Employee.DepartmentId FK points to Department.Id

4.1 Justification and Alternatives Analysis

Some alternatives when designing the tables and data types are briefly discussed here and the reasons for choosing the current design over these alternatives are justified. This section primarily focuses on the data types chosen for table properties – for justification and alternatives analysis for the entity structure please see the ER diagram analysis section.

4.1.1 Enumeration Data Type

An enumeration data type was used to represent the various enumeration-like properties on an entity (such as Employee.Role). This was done as a convenient way to both restrict user input to the allowed value set and to reduce the size of the data field (given that enumerations have a known length they can be stored more efficiently than conventional varchar strings).

Another alternative could be to use a check constraint however, although there is some question if this is supported in MySQL current [2], or a trigger to validate the input however this introduces additional complexity. A third alternative would be to reference a separate table containing the enumeration values by foreign key and validate the user input in this way however this again introduces more complexity in the form of additional tables and joins.

4.1.2 Decimal Type to Represent Money

A decimal type was used to represent money values. This is the recommended type when working with money according to the MySQL guidelines and avoids data loss due to the representation of 32 or 64 bit floating point numbers [3].

4.1.3 Phone Field Representation

Since a 10 digit phone number format is assumed (including the area code) a BIGINT was required given that an INT of 32 bits will not be able to represent the full length of a 10 digit phone number. A width value of 10 was also put (BIGINT(10)).

5 Scripts



If the database is not yet selected, please execute the `USE <DatabaseName>;` command prior to executing scripts.

5.1 Creation Script

The script creates the tables and data types representing the ER diagram and database diagram in Figure 1 and Figure 2 respectively.

Listing 1 – Table Creation script

```
CREATE TABLE Department (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL
);

CREATE TABLE Employee (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL,
  DepartmentId INT NOT NULL,
  Role ENUM ('Manager', 'Sales Associate', 'Developer') NOT NULL,
  ManagerId INT,

  FOREIGN KEY (DepartmentId) REFERENCES Department (Id),
  FOREIGN KEY (ManagerId) REFERENCES Employee (Id)
  ON DELETE SET NULL
);

CREATE INDEX idx_employee_role
ON Employee (Role);

CREATE TABLE Company (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(256) NOT NULL,
  Address VARCHAR(256) NOT NULL,
  ContactFirstName VARCHAR(256) NOT NULL,
  ContactLastName VARCHAR(256) NOT NULL,
  ContactNumber BIGINT(10) NOT NULL,
  ContactEmail VARCHAR(256) NOT NULL
);

CREATE TABLE Contract (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  Type ENUM ('Premium', 'Gold', 'Diamond', 'Silver') NOT NULL,
  CompanyId INT NOT NULL,
  AnnualContractValue DECIMAL(20, 2) NOT NULL,
  InitialAmount DECIMAL(20, 2) NOT NULL,
  StartDate DATETIME NOT NULL DEFAULT NOW(),
  Service ENUM ('Cloud', 'On-premises') NOT NULL,

  FOREIGN KEY (CompanyId) REFERENCES Company (Id)
);

CREATE INDEX idx_contract_type
ON Contract (Type);
CREATE INDEX idx_contract_service
ON Contract (Service);
```

```

CREATE TABLE WorksOn (
    EmployeeId INT,
    ContractId INT,

    CONSTRAINT WorksOn_pk PRIMARY KEY (EmployeeId, ContractId),
    CONSTRAINT chk_EmployeeID FOREIGN KEY (EmployeeId) REFERENCES Employee (Id) ON DELETE
    CASCADE,
    CONSTRAINT chk_ContractID FOREIGN KEY (ContractId) REFERENCES Contract (Id) ON DELETE CASCADE
);

```

5.2 Insert Data Script

The insertion script inserts:

- 76 employee records (composed of 6 managers, 10 sales associates and 60 developers)
- 6 departments
- 12 contracts
- 6 companies

Employees are uniformly distributed among departments, managers and contracts.

Listing 2 – Mock data insertion script

```

INSERT INTO Department (Id, Name)
VALUES (1, 'Development'), (2, 'QA'), (3, 'UI'), (4, 'Design'), (5, 'Business Intelligence'),
(6, 'Networking');

INSERT INTO Employee (Id, Role, Name, DepartmentId, ManagerId)
VALUES (1, 'Manager', 'Van David', 1, null), (2, 'Manager', 'Reta Bayer', 2, null),
(3, 'Manager', 'Tyreek Cole', 3, null), (4, 'Manager', 'David', 4, null),
(5, 'Manager', 'Jonatan Batz', 5, null), (6, 'Manager', 'Armani Waters', 6, null),
(7, 'Sales Associate', 'Cordia Cummerata', 1, 1), (8, 'Sales Associate', 'Lenore Smitham', 2,
2),
(9, 'Sales Associate', 'Nikko Hintz', 3, 3), (10, 'Sales Associate', 'Domenica Ankunding', 4,
4),
(11, 'Sales Associate', 'Camron Schaden', 5, 5), (12, 'Sales Associate', 'Lesly Turcotte', 6,
6),
(13, 'Sales Associate', 'Cecile O'Kon', 1, 1), (14, 'Sales Associate', 'Freddie Kihn', 2,
2),
(15, 'Sales Associate', 'Bradley Bayer', 3, 3), (16, 'Sales Associate', 'Preston Balistreri',
4, 4),
(17, 'Developer', 'Addison Douglas', 1, 1), (18, 'Developer', 'Myrtis Koepp', 2, 2),
(19, 'Developer', 'Stephon Greenholt', 3, 3), (20, 'Developer', 'Josh Fetcher', 4, 4),
(21, 'Developer', 'Laisha Collier', 5, 5), (22, 'Developer', 'Fidel Heller', 6, 6),
(23, 'Developer', 'Clarissa Gaylord', 1, 1), (24, 'Developer', 'Manuela Beahan', 2, 2),
(25, 'Developer', 'Maribel Kautzer', 3, 3), (26, 'Developer', 'Maiya ammes', 4, 4),
(27, 'Developer', 'Oleta Toy', 5, 5), (28, 'Developer', 'Chaim Daugherty', 6, 6),
(29, 'Developer', 'Ryann Gutmann', 1, 1), (30, 'Developer', 'Schuyler Schmeler', 2, 2),
(31, 'Developer', 'Mabelle Hermiston', 3, 3), (32, 'Developer', 'Edna Daniel', 4, 4),
(33, 'Developer', 'Noble Roob', 5, 5), (34, 'Developer', 'Jammie Feil', 6, 6),
(35, 'Developer', 'Rey Eichmann', 1, 1), (36, 'Developer', 'Rachael Kuvalis', 2, 2),
(37, 'Developer', 'Wade Breitenberg', 3, 3), (38, 'Developer', 'Larue Hilpert', 4, 4),
(39, 'Developer', 'Mina Ondricka', 5, 5), (40, 'Developer', 'Misty Konopelski', 6, 6),
(41, 'Developer', 'Lillian Witting', 1, 1), (42, 'Developer', 'Annetta Spinka', 2, 2),
(43, 'Developer', 'Abigail Glover', 3, 3), (44, 'Developer', 'Shanny Abshire', 4, 4),
(45, 'Developer', 'Haylee Hyatt', 5, 5), (46, 'Developer', 'Gustave Mraz', 6, 6),
(47, 'Developer', 'Enrique Kuhlman', 1, 1), (48, 'Developer', 'Drake Jenkins', 2, 2),
(49, 'Developer', 'Dominique Glover', 3, 3), (50, 'Developer', 'Everardo Gutmann', 4, 4),
(51, 'Developer', 'Schuyler Rau', 5, 5), (52, 'Developer', 'Mose Kemmer', 6, 6),
(53, 'Developer', 'Alysa Murray', 1, 1), (54, 'Developer', 'Rodolfo Labadie', 2, 2),

```

```

(55, 'Developer', 'Arden Maggio', 3, 3), (56, 'Developer', 'Giles Murazik', 4, 4),
(57, 'Developer', 'Marty Wisozk', 5, 5), (58, 'Developer', 'Alexandrine Goldner', 6, 6),
(59, 'Developer', 'Carmine Maggio', 1, 1), (60, 'Developer', 'Orlando Dooley', 2, 2),
(61, 'Developer', 'Judge Johnson', 3, 3), (62, 'Developer', 'Mikel Schumm', 4, 4),
(63, 'Developer', 'Eriberto Johnston', 5, 5), (64, 'Developer', 'Crawford Bogan', 6, 6),
(65, 'Developer', 'Hector Gutmann', 1, 1), (66, 'Developer', 'Joannie McDermott', 2, 2),
(67, 'Developer', 'Jordan Toy', 3, 3), (68, 'Developer', 'Anya Windler', 4, 4),
(69, 'Developer', 'Christine Hirthe', 5, 5), (70, 'Developer', 'Priscilla Mertz', 6, 6),
(71, 'Developer', 'Kole Metz', 1, 1), (72, 'Developer', 'Wendell Fadel', 2, 2),
(73, 'Developer', 'Cordelia Hamill', 3, 3), (74, 'Developer', 'Felix Gislason', 4, 4),
(75, 'Developer', 'Frederic Corkery', 5, 5), (76, 'Developer', 'Taylor Paucek', 6, 6);

INSERT INTO Company (Id, Name, Address, ContactFirstName, ContactLastName, ContactNumber,
ContactEmail) VALUES
(1, 'CSC Corp.', '62 Kent St E Lindsay ON K9V 2C5', 'Adolf', 'Denesik', 3915280674,
'adolf.denesik@csccorp.biz'),
(2, 'Heavy Industries', '4520 7th Ave W Vancouver BC V6R 1X3', 'Abigayle', 'Gibson',
7290345264,
'abigayle.gibson@hv.org'),
(3, 'Seacoast Software', '4915 52 St , Bruderheim, AB, T0B 0S0', 'Bonita', 'Zulauf',
1721138850,
'bonita.zulauf@seacoast.ca'),
(4, 'Canopy University', '54 Rue Du Calvados , Candiac, QC, J5R 6H4', 'Shanelle', 'Keeling',
1880369913,
'shanelle.keeling@canopy.edu'),
(5, 'Coin Stomp', '2013 Manston Rd , Black Creek, BC, V9J 1A6', 'Estevan', 'Effertz',
1994485120,
'estevan.effertz@coinstomp.org'),
(6, 'Obverso', '5411 Lakeshore Rd 14 Burlington ON L7L 1E1', 'Darryl', 'Bins', 1886942583,
'darryl.bins@obverso.biz');

INSERT INTO Contract (Id, Type, CompanyId, AnnualContractValue, InitialAmount, Service,
StartDate)
VALUES (1, 'Gold', 1, 17379.200000, 1639.280000, 'Cloud', '2016-01-22T00:00'),
(2, 'Silver', 2, 12331.690000, 1497.250000, 'On-premises', '2017-01-20T00:00'),
(3, 'Premium', 3, 32891.520000, 1629.240000, 'Cloud', '2016-10-28T00:00'),
(4, 'Diamond', 4, 33067.560000, 560.400000, 'On-premises', '2017-08-18T00:00'),
(5, 'Gold', 5, 15630.850000, 1211.320000, 'Cloud', '2017-03-31T00:00'),
(6, 'Silver', 6, 25784.810000, 558.490000, 'On-premises', '2016-07-07T00:00'),
(7, 'Premium', 3, 36714.830000, 1094.250000, 'Cloud', '2017-04-23T00:00'),
(8, 'Diamond', 2, 26052.730000, 1327.850000, 'On-premises', '2017-07-28T00:00'),
(9, 'Gold', 3, 28340.080000, 1792.220000, 'Cloud', '2017-06-07T00:00'),
(10, 'Silver', 4, 15305.270000, 1388.300000, 'On-premises', '2016-08-26T00:00'),
(11, 'Premium', 5, 31836.630000, 682.270000, 'Cloud', '2017-05-19T00:00'),
(12, 'Diamond', 6, 27813.110000, 1968.620000, 'On-premises', '2017-10-13T00:00');

INSERT INTO WorksOn (EmployeeId, ContractId)
VALUES (5, 1), (8, 1), (76, 1), (52, 1), (71, 1), (49, 1), (54, 1), (2, 2), (7, 2), (53, 2),
(72, 2), (26, 2), (29, 2),
(68, 2), (1, 3), (14, 3), (23, 3), (56, 3), (69, 3), (35, 3), (42, 3), (6, 4), (12, 4), (51,
4), (61, 4), (47, 4),
(21, 4), (30, 4), (27, 4), (59, 4), (3, 5), (9, 5), (50, 5), (18, 5), (65, 5), (43, 5), (63,
5), (4, 6), (13, 6),
(67, 6), (31, 6), (73, 6), (19, 6), (33, 6), (5, 7), (16, 7), (55, 7), (34, 7), (44, 7), (32,
7), (24, 7), (2, 8),
(10, 8), (74, 8), (25, 8), (28, 8), (22, 8), (38, 8), (1, 9), (11, 9), (36, 9), (48, 9), (20,
9), (58, 9), (57, 9),
(17, 9), (62, 9), (41, 9), (6, 10), (15, 10), (64, 10), (46, 10), (70, 10), (39, 10), (75,
10), (66, 10), (40, 10),
(3, 11), (8, 11), (45, 11), (60, 11), (37, 11), (76, 11), (52, 11), (4, 12), (7, 12), (71,
12), (49, 12), (54, 12),
(53, 12), (72, 12), (26, 12);

```

5.3 Queries

The sample SQL queries done for this submission found are briefly reviewed here. Printouts of the results of each query can be found in the appendix section as this section focuses more on the analysis of each query.

5.3.1 List of Employees Working Exclusively on Cloud Contracts

By “exclusively”, we interpret this to mean that employees are working solely on cloud contracts and no other type.

To complete this query, we look at the list of all employees working on contracts and exclude all ids of employees who are working on at least one non-cloud contract (the inner subquery) The result are employees working exclusively on cloud contracts.

```
SELECT DISTINCT E.Id, E.Name, E.Role
FROM Employee E
  INNER JOIN WorksOn WO ON WO.EmployeeId = E.Id
WHERE WO.EmployeeId NOT IN (SELECT DISTINCT WO.EmployeeId
                             FROM WorksOn WO
                             INNER JOIN Contract Co ON Co.Id = WO.ContractId
                             WHERE Co.Service != 'Cloud') ORDER BY E.Id;
```

5.3.2 Employee Ids of all developers

A simple role-based query is done to retrieve the desired results. We select Name and Role for convenience in addition to showing all id fields.

```
SELECT Id, Name, Role
FROM Employee
WHERE Role = 'Developer';
```

5.3.3 Employees Working on Contract with ACV of at least \$20,000

A join between the contracts and employees table is done with a condition of the annual contract value being at least \$20,000 is applied.

```
SELECT DISTINCT E.Id, E.Name, E.Role
FROM Employee E
  INNER JOIN WorksOn WO ON WO.EmployeeId = E.Id
  INNER JOIN Contract Co ON Co.Id = WO.ContractId
WHERE Co.AnnualContractValue >= 20000;
```

5.3.4 Contracts Managed by Van David

A join between the contracts and employees table is done and those contracts with Van David as manager are selected. Since it is already known that Van David is a manager no WHERE Role = 'Manager' check is required.

```
SELECT E.Name AS Manager, Co.*
FROM WorksOn WO
  INNER JOIN Contract Co ON Co.Id = WO.ContractId
  INNER JOIN Employee E ON WO.EmployeeId = E.Id
WHERE Name = 'Van David';
```

6 Assumptions

The following assumptions were made during the design and implementation:

- Employees have a single job / role that remains the same for all contracts. It is impossible that an employee works as a manager for one contract and as a Developer for another.
- Managers work as managers of contracts and do not complete any other role. For example, a Manager may not work as a Developer on a contract.
- The company / client has a single person of contact that remains the same for all contracts. Hence contact information is put as a property of the company.
- The “responsible of the company” listed in the warm-up project instructions [1] is assumed to be the company’s person of contact to which the contact number and company email are associated.
- A department may have 1 or many managers associated with it
- We assume a phone format of 10 digits (including the area code) with no dash or period separators when inputting a phone number. For example: 5145551111.

7 Distribution of Work

Table 4 – Distribution of work among team members

Task	Team Member(s)
ER diagram	Faezeh Mobasheri (26821022) Jasdeep Ratot (26383599) Benardo Sandi Morales (40046377) Aida Sharif Rohani (21341669) Stefan Wapnick (26211798) <i>(Done jointly during team meeting)</i>
Creation script	Faezeh Mobasheri (26383599) Benardo Sandi Morales (40046377)
Insert data script and data generation	Stefan Wapnick (26211798)
Queries script	Benardo Sandi Morales (40046377) Aida Sharif Rohani (21341669)
Report documentation	Jasdeep Ratot (26383599) Stefan Wapnick (26211798)

References

- [1] K. Jababo, “COMP 5531 – Files and Databases – Summer 2018 – WarmUp Project”, Concordia Univeristy, 2018. Available: https://moodle.concordia.ca/moodle/pluginfile.php/3181015/mod_resource/content/1/COMP5531-Summer-2018-WarmUp-Project.pdf [Accessed: 2018-06-17]
- [2] G. Humphries. “Check constraints not supported”, MySQL Feature Request Forums. 2013 Available: <https://bugs.mysql.com/bug.php?id=69556> [Accessed 2018-06-17]
- [3] N.A. “Fixed-Point Types (Exact Value) – DECIMAL, NUMERIC”, MySQL Documentation. 2018. Available: <https://dev.mysql.com/doc/refman/5.7/en/fixed-point-types.html> [Accessed 2018-06-17]

Appendix A –Tables Data Printouts

This section includes printouts of the mock data seeded in the database tables

Table 5 – Data inserted into *Department* table

Id	Name
1	Development
2	QA
3	UI
4	Design
5	Business Intelligence
6	Networking

Table 6 – Data inserted into *Employee* table

Id	Name	DepartmentId	Role	ManagerId
1	Van David	1	Manager	null
2	Reta Bayer	2	Manager	null
3	Tyreek Cole	3	Manager	null
4	David	4	Manager	null
5	Jonatan Batz	5	Manager	null
6	Armani Waters	6	Manager	null
7	Cordia Cummerata	1	Sales Associate	1
8	Lenore Smitham	2	Sales Associate	2
9	Nikko Hintz	3	Sales Associate	3
10	Domenica Ankunding	4	Sales Associate	4
11	Camron Schaden	5	Sales Associate	5
12	Lesly Turcotte	6	Sales Associate	6
13	Cecile O'Kon	1	Sales Associate	1
14	Freddie Kihn	2	Sales Associate	2
15	Bradley Bayer	3	Sales Associate	3
16	Preston Balistreri	4	Sales Associate	4
17	Addison Douglas	1	Developer	1
18	Myrtis Koepp	2	Developer	2
19	Stephon Greenholt	3	Developer	3
20	Josh Fetcher	4	Developer	4
21	Laisha Collier	5	Developer	5
22	Fidel Heller	6	Developer	6
23	Clarissa Gaylord	1	Developer	1
24	Manuela Beahan	2	Developer	2
25	Maribel Kautzer	3	Developer	3
26	Maiya ammes	4	Developer	4
27	Oleta Toy	5	Developer	5
28	Chaim Daugherty	6	Developer	6
29	Ryann Gutmann	1	Developer	1
30	Schuyler Schmeler	2	Developer	2
31	Mabelle Hermiston	3	Developer	3
32	Edna Daniel	4	Developer	4
33	Noble Roob	5	Developer	5
34	Jammie Feil	6	Developer	6

35	Rey Eichmann	1	Developer	1
36	Rachael Kuvalis	2	Developer	2
37	Wade Breitenberg	3	Developer	3
38	Larue Hilpert	4	Developer	4
39	Mina Ondricka	5	Developer	5
40	Misty Konopelski	6	Developer	6
41	Lilian Witting	1	Developer	1
42	Annetta Spinka	2	Developer	2
43	Abigail Glover	3	Developer	3
44	Shanny Abshire	4	Developer	4
45	Haylee Hyatt	5	Developer	5
46	Gustave Mraz	6	Developer	6
47	Enrique Kuhlman	1	Developer	1
48	Drake Jenkins	2	Developer	2
49	Dominique Glover	3	Developer	3
50	Everardo Gutmann	4	Developer	4
51	Schuyler Rau	5	Developer	5
52	Mose Kemmer	6	Developer	6
53	Alysa Murray	1	Developer	1
54	Rodolfo Labadie	2	Developer	2
55	Arden Maggio	3	Developer	3
56	Giles Murazik	4	Developer	4
57	Marty Wisozk	5	Developer	5
58	Alexandrine Goldner	6	Developer	6
59	Carmine Maggio	1	Developer	1
60	Orlando Dooley	2	Developer	2
61	Judge Johnson	3	Developer	3
62	Mikel Schumm	4	Developer	4
63	Eriberto Johnston	5	Developer	5
64	Crawford Bogan	6	Developer	6
65	Hector Gutmann	1	Developer	1
66	Joannie McDermott	2	Developer	2
67	Jordan Toy	3	Developer	3
68	Anya Windler	4	Developer	4
69	Christine Hirthe	5	Developer	5
70	Priscilla Mertz	6	Developer	6
71	Kole Metz	1	Developer	1
72	Wendell Fadel	2	Developer	2
73	Cordelia Hamill	3	Developer	3
74	Felix Gislason	4	Developer	4
75	Frederic Corkery	5	Developer	5
76	Taylor Paucek	6	Developer	6

Table 7 – Data inserted into *Company* table

Id	Name	Address	ContactFirst Name	ContactLast Name	ContactNumber	ContactEmail
1	CSC Corp.	62 Kent St E Lindsay ON K9V 2C5	Adolf	Denesik	3915280674	adolf.denesik@csccorp.biz
2	Heavy Industries	4520 7th Ave W Vancouver BC V6R 1X3	Abigayle	Gibson	7290345264	abigayle.gibson@hv.org

3	Seacoast Software	4915 52 St , Bruderheim, AB, T0B 0S0	Bonita	Zulauf	1721138850	bonita.zulauf@seacost.ca
4	Canopy University	54 Rue Du Calvados , Candiac, QC, J5R 6H4	Shanelle	Keeling	1880369913	shanelle.keeling@canopy.edu
5	Coin Stomp	2013 Manston Rd , Black Creek, BC, V9J 1A6	Estevan	Effertz	1994485120	estevan.effertz@coinstomp.org
6	Obverso	5411 Lakeshore Rd 14 Burlington ON L7L 1E1	Darryl	Bins	1886942583	darryl.bins@obverso.biz

Table 8 – Data inserted into **Contract** table

Id	Type	CompanyId	AnnualContractValue	InitialAmount	StartDate	Service
1	Gold	1	17379.2	1639.28	2016-01-22 0:00	Cloud
2	Silver	2	12331.69	1497.25	2017-01-20 0:00	On-premises
3	Premium	3	32891.52	1629.24	2016-10-28 0:00	Cloud
4	Diamond	4	33067.56	560.4	2017-08-18 0:00	On-premises
5	Gold	5	15630.85	1211.32	2017-03-31 0:00	Cloud
6	Silver	6	25784.81	558.49	2016-07-07 0:00	On-premises
7	Premium	3	36714.83	1094.25	2017-04-23 0:00	Cloud
8	Diamond	2	26052.73	1327.85	2017-07-28 0:00	On-premises
9	Gold	3	28340.08	1792.22	2017-06-07 0:00	Cloud
10	Silver	4	15305.27	1388.3	2016-08-26 0:00	On-premises
11	Premium	5	31836.63	682.27	2017-05-19 0:00	Cloud
12	Diamond	6	27813.11	1968.62	2017-10-13 0:00	On-premises

Table 9 – Data inserted into **WorksOn** table (table is continued horizontally to save space)

EmployeeId	ContractId	EmployeeId	ContractId	EmployeeId	ContractId	EmployeeId	ContractId
5	1	21	4	24	7	15	10
8	1	27	4	32	7	39	10
49	1	30	4	34	7	40	10
52	1	47	4	44	7	46	10
54	1	51	4	55	7	64	10
71	1	59	4	2	8	66	10
76	1	61	4	10	8	70	10
2	2	3	5	22	8	75	10
7	2	9	5	25	8	3	11
26	2	18	5	28	8	8	11
29	2	43	5	38	8	37	11
53	2	50	5	74	8	45	11
68	2	63	5	1	9	52	11
72	2	65	5	11	9	60	11
1	3	4	6	17	9	76	11
14	3	13	6	20	9	4	12
23	3	19	6	36	9	7	12
35	3	31	6	41	9	26	12
42	3	33	6	48	9	49	12
56	3	67	6	57	9	53	12
69	3	73	6	58	9	54	12

6	4
12	4

5	7
16	7

62	9
6	10

71	12
72	12

Appendix B – Query Results Printouts

Table 10 – *Question 1 Results: Employees working exclusively on cloud*

Id	Name	Role
1	Van David	Manager
3	Tyreek Cole	Manager
5	Jonatan Batz	Manager
8	Lenore Smitham	Sales Associate
9	Nikko Hintz	Sales Associate
11	Camron Schaden	Sales Associate
14	Freddie Kihn	Sales Associate
16	Preston Balistreri	Sales Associate
17	Addison Douglas	Developer
18	Myrtis Koepp	Developer
20	Josh Fetcher	Developer
23	Clarissa Gaylord	Developer
24	Manuela Beahan	Developer
32	Edna Daniel	Developer
34	Jammie Feil	Developer
35	Rey Eichmann	Developer
36	Rachael Kuvalis	Developer
37	Wade Breitenberg	Developer
41	Lilian Witting	Developer
42	Annetta Spinka	Developer
43	Abigail Glover	Developer
44	Shanny Abshire	Developer
45	Haylee Hyatt	Developer
48	Drake Jenkins	Developer
50	Everardo Gutmann	Developer
52	Mose Kemmer	Developer
55	Arden Maggio	Developer
56	Giles Murazik	Developer
57	Marty Wisozk	Developer
58	Alexandrine Goldner	Developer
60	Orlando Dooley	Developer
62	Mikel Schumm	Developer
63	Eriberto Johnston	Developer
65	Hector Gutmann	Developer
69	Christine Hirthe	Developer
76	Taylor Paucek	Developer

Table 11 – *Question 2 Results: List of developers*

Id	Name	Role
17	Addison Douglas	Developer
18	Myrtis Koepp	Developer
19	Stephon Greenholt	Developer
20	Josh Fetcher	Developer
21	Laisha Collier	Developer
22	Fidel Heller	Developer

23	Clarissa Gaylord	Developer
24	Manuela Beahan	Developer
25	Maribel Kautzer	Developer
26	Maiya ammes	Developer
27	Oleta Toy	Developer
28	Chaim Daugherty	Developer
29	Ryann Gutmann	Developer
30	Schuyler Schmeler	Developer
31	Mabelle Hermiston	Developer
32	Edna Daniel	Developer
33	Noble Roob	Developer
34	Jammie Feil	Developer
35	Rey Eichmann	Developer
36	Rachael Kuvalis	Developer
37	Wade Breitenberg	Developer
38	Larue Hilpert	Developer
39	Mina Ondricka	Developer
40	Misty Konopelski	Developer
41	Lilian Witting	Developer
42	Annetta Spinka	Developer
43	Abigail Glover	Developer
44	Shanny Abshire	Developer
45	Haylee Hyatt	Developer
46	Gustave Mraz	Developer
47	Enrique Kuhlman	Developer
48	Drake Jenkins	Developer
49	Dominique Glover	Developer
50	Everardo Gutmann	Developer
51	Schuyler Rau	Developer
52	Mose Kemmer	Developer
53	Alysa Murray	Developer
54	Rodolfo Labadie	Developer
55	Arden Maggio	Developer
56	Giles Murazik	Developer
57	Marty Wisozk	Developer
58	Alexandrine Goldner	Developer
59	Carmine Maggio	Developer
60	Orlando Dooley	Developer
61	Judge Johnson	Developer
62	Mikel Schumm	Developer
63	Eriberto Johnston	Developer
64	Crawford Bogan	Developer
65	Hector Gutmann	Developer
66	Joannie McDermott	Developer
67	Jordan Toy	Developer
68	Anya Windler	Developer
69	Christine Hirthe	Developer
70	Priscilla Mertz	Developer
71	Kole Metz	Developer
72	Wendell Fadel	Developer
73	Cordelia Hamill	Developer
74	Felix Gislason	Developer
75	Frederic Corkery	Developer
76	Taylor Paucek	Developer

Table 12 – Question 3: Employees working on contracts with ACV >= \$20,000

Id	Name	Role
1	Van David	Manager
14	Freddie Kihn	Sales Associate
23	Clarissa Gaylord	Developer
35	Rey Eichmann	Developer
42	Annetta Spinka	Developer
56	Giles Murazik	Developer
69	Christine Hirthe	Developer
6	Armani Waters	Manager
12	Lesly Turcotte	Sales Associate
21	Laisha Collier	Developer
27	Oleta Toy	Developer
30	Schuyler Schmeler	Developer
47	Enrique Kuhlman	Developer
51	Schuyler Rau	Developer
59	Carmine Maggio	Developer
61	Judge Johnson	Developer
4	David	Manager
13	Cecile O'Kon	Sales Associate
19	Stephon Greenholt	Developer
31	Mabelle Hermiston	Developer
33	Noble Roob	Developer
67	Jordan Toy	Developer
73	Cordelia Hamill	Developer
5	Jonatan Batz	Manager
16	Preston Balistreri	Sales Associate
24	Manuela Beahan	Developer
32	Edna Daniel	Developer
34	Jammie Feil	Developer
44	Shanny Abshire	Developer
55	Arden Maggio	Developer
2	Reta Bayer	Manager
10	Domenica Ankunding	Sales Associate
22	Fidel Heller	Developer
25	Maribel Kautzer	Developer
28	Chaim Daugherty	Developer
38	Larue Hilpert	Developer
74	Felix Gislason	Developer
11	Camron Schaden	Sales Associate
17	Addison Douglas	Developer
20	Josh Fetcher	Developer
36	Rachael Kuvalis	Developer
41	Lilian Witting	Developer
48	Drake Jenkins	Developer
57	Marty Wisozk	Developer
58	Alexandrine Goldner	Developer
62	Mikel Schumm	Developer
3	Tyreek Cole	Manager
8	Lenore Smitham	Sales Associate
37	Wade Breitenberg	Developer
45	Haylee Hyatt	Developer
52	Mose Kemmer	Developer
60	Orlando Dooley	Developer
76	Taylor Paucek	Developer
7	Cordia Cummerata	Sales Associate
26	Maiya ammes	Developer
49	Dominique Glover	Developer

53	Alysa Murray	Developer
54	Rodolfo Labadie	Developer
71	Kole Metz	Developer
72	Wendell Fadel	Developer

Table 13 – **Question 4:** Contracts managed by Van David

Manager	ContractId	Type	CompanyId	AnnualContractValue	InitialAmount	StartDate	Service
Van David	3	Premium	3	32891.52	1629.24	2016-10-28 00:00:00	Cloud
Van David	9	Gold	3	28340.08	1792.22	2017-06-07 00:00:00	Cloud