**Construction of Data Base for Ice Cream Store Client: Scoop There It Is**

By Graham Novitch, Wendy Ralston, Brandon Vasques, and Heather Zeluff

**I. Business Scenario**

The following models, normalization, SQL code, etc. are comprehensive building blocks in database design and development for a scheduling app. This platform will provide a much-needed service for local ice cream business “Scoop There It Is” based in New Mexico. The database's primary objective is to improve the coordination of ice cream deliveries from the supplier, "Ice Ice Baby," to the store location, considering the state's high temperatures and the need to prevent the ice cream from melting.

The priority of this database will better match the deliveries of ice cream from the supplier, “Ice Ice Baby” to the number of employees at the store location. Given the state’s high temperatures, the ice cream needs to be loaded from the delivery truck into the store’s freezers before melting. This scheduling app will ensure that the store is staffed properly to accomplish this task.

Secondly, this app will also schedule the ideal delivery for the ice cream to the store. The database addressed the need for proper staffing to unload delivery trucks and manage the ice cream in the store's freezers, as well as scheduling ideal delivery times based on weather conditions. It will compare typical weather conditions for the time of year to the time of day it should arrive. For example, in July, it should be dropped off as early or as late as possible to avoid the heat of the heat of the day. Lastly, the scheduling app will enable management to track the temperature of the ice cream to ensure quality.

For the ease of our reader, Scoop There It Is will be abbreviated as STII, and Ice Ice Baby will be abbreviated as IIB throughout our project and the models.

**II. ER Model using UML Notation**

Based on our above database requirements, we created this Entity Relationship model using UML notation.

**Commentary on ER Model:**

* The entities are arranged in relative left to right order from supplier, to order fulfillment, to ice cream storage. This allows us to easily view the overall fulfillment process and the entities involved in each step.
* No relationship lines cross, making this model easy to view and understand.
* Attribute names use no spaces and abbreviations are used for viewing ease, as all database users will be intimately familiar with the two companies and their processes.
* Because this is the first stab at a database connecting these two companies, we have added a list of assumptions about the two companies for clarity.
* The diagram also has a legend which titles the project and provides information about its creation and updates.
* We addressed the many-to-many multiplicity relationship between STII Managers and STII Employee, creating an intersect table.
* Our model only contains one composite key: a combination of the two primary keys in the STII\_Managers\_STII\_Employee intersect table. Composite keys were not otherwise needed because all our primary keys were unique to each entity.

**Relationship Sentences**

One IIB **Truck Driver**, one **IIB Truck**, and one **NM Live Weather Condition** *must* enable at least one **IIB Delivery Schedule**.

One **IIB Delivery Schedule** *must* be determined by at least one **Order from STII to IIB** placed by at least one **STII Manager**, which *must be* sent back to the **STII Manager**.

At least one **STII Managers** *must* create one **STII Staffing Schedule**, manage at least 1 **STII Employee** and oversee one **STII Freezer** each.

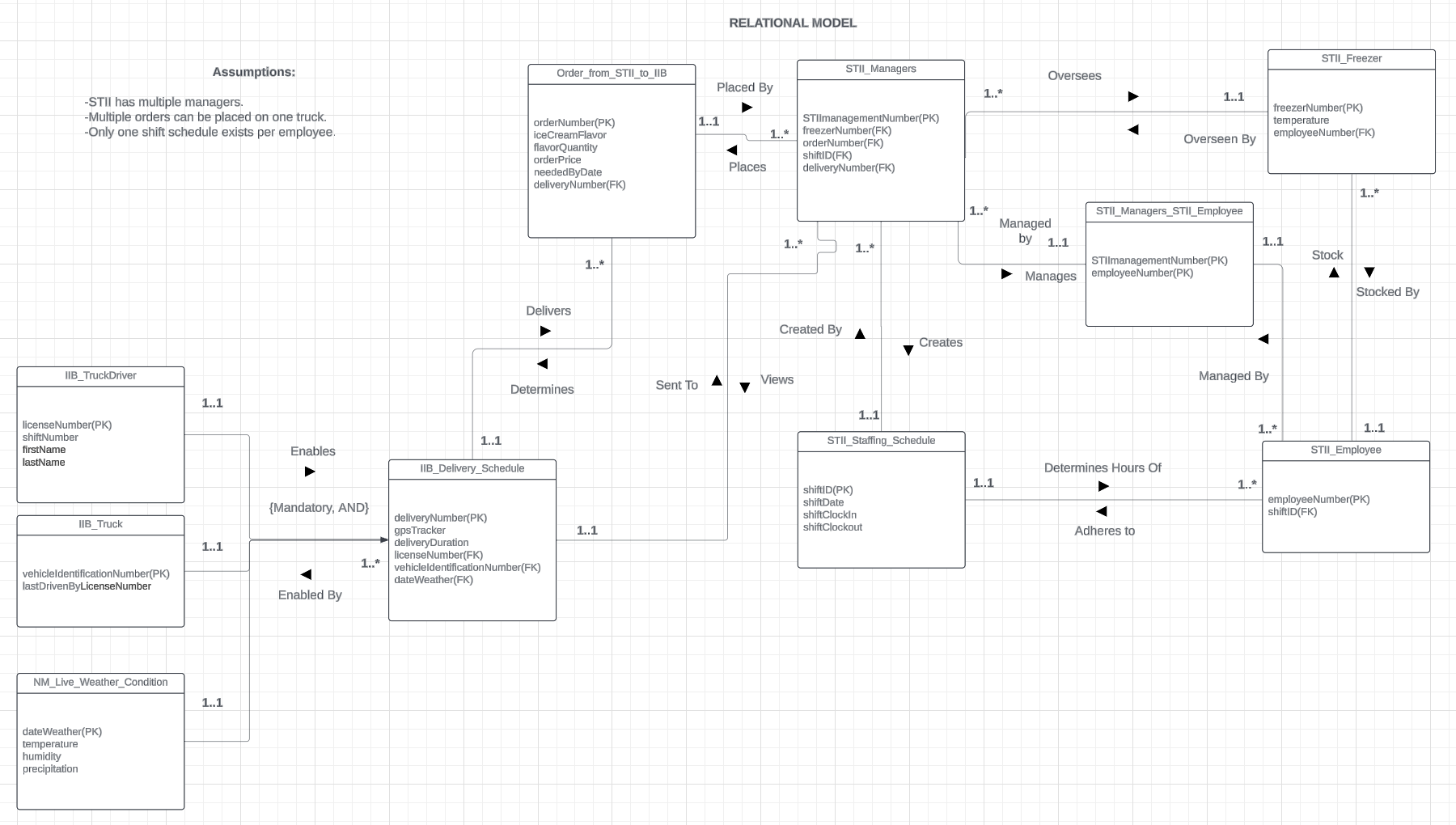
One **STII Staffing Schedule** *must* determine the hours of at least one **STII Employees**.

One **STII Employee** *must* stock at least one **STII Freezer**.

**Commentary on Relationship Sentences**:

* The database entities, relationships, and multiplicities are written word for word to further explain our ER model.
* The relationship phrases are underlined. The entity names are bolded. The cardinality is in italics. The multiplicity is written as “one” or “at least one”.

**III. Conversion to Relational Model**



After initially creating the ER model, we converted it to a relational model. We added foreign keys to the necessary entities. Below are the related entities with the added foreign keys. The primary keys are marked “PK” while the foreign keys are marked “FK.” These are the relationships as they stand at this step. There are no duplicate attributes other than foreign keys. This is because the entities are vastly differ from each other in function within this scheduling app.

1) **IIB\_TruckDriver** (licenseNumber(PK), shiftNumber, firstName, LastName)

2) **IIB\_Truck** (vehicleIdentificationNumber(PK), lastDrivenByLicenseNumber)

3) **NM\_Live\_Weather\_Condition** (dateWeather(PK), temperature, humidity, precipitation)

4) **IIB\_Delivery\_Schedule** (deliveryNumber (PK), gpsTracker, deliveryDuration)

5) **Order\_from\_STII\_to\_IIB** (orderNumber(PK), iceCreamFlavor(PK), flavorQuantity, flavorPrice, neededByDate, deliveryNumber(FK))

6) **STII\_Managers** (STIImanagementNumber(PK), freezerNumber(FK), orderNumber(FK), shiftID(FK), employeeNumber(FK), deliveryNumber(FK))

7) STII\_Managers\_STII\_Employee (STIImanagementNumber(PK), employeeNumber(PK))

8) **STII\_Staffing\_Schedule** (shiftID(PK), shiftDate, shiftClockIn, ShiftClockout)

9) **STII\_Employee (**employeeNumber(PK), shiftID(FK))

10) **STII\_Freezer** (freezerNumber(PK), temperature, employeeNumber(FK))

**Commentary on Conversion to Relational Model:**

* Notice there is a composite key under STII\_Managers\_STII\_Employee.
* Notice there is a composite key under Order\_from\_STII\_to\_IIB, as there can be more than one flavor per order number.

**IV. Normalization**

Next, we normalized the relations defined by the relational model.

1) **IIB\_TruckDriver** (licenseNumber(PK), shiftNumber, firstName, LastName)

Sample Data

|  |  |  |  |
| --- | --- | --- | --- |
| licenseNumber (PK) | shiftNumber | firstName | lastName |
| NM 294587-09 | 45879 | William | Erickson |
| NM 965489-02 | 45880 | Kent | Davidson |
| NM 145286-78 | 45881 | Amy | Brookheimer |
| NM 064023-12 | 45882 | Marjorie | Palmiotti |
| NM 770290-12 | 45883 | Jonah | Ryan |
| NM 348206-97 | 45884 | Catherine | Meyer |
| NM 082597-20 | 45885 | Roger | Furlong |
| NM 503678-99 | 45886 | Susan | Wilson |

Key: licenseNumber

FD1: licenseNumber -> shiftNumber, firstName, lastName

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

2) IIB\_Truck (vehicleIdentificationNumber (PK), lastDrivenByLicenseNumber)

Sample Data

|  |  |  |
| --- | --- | --- |
| vehicleIdentificationNumber (PK) | lastDrivenByLicenseNumber |  |
| 4Y1SL65848Z411439 | NM 294587-09 |  |
| 4M2EL65648T444468 | NM 065489-02 |  |
| 5H2RK789648T12446 | NM 145286-78 |  |
| 1T1QL12848N400032 | NM 064023-12 |  |
| 274NME3057THDF34 | NM 770290-12 |  |
| 283WUVH3489EUG6 | NM 348206-97 |  |
| 978UOI3850NMN593 | NM 082597-20 |  |
| 364HSN4086NRI3334 | NM 503678-99 |  |

Key: vehicleIdentificationNumber

FD1: vehicleIdentificationNumber -> lastDrivenByLicenseNumber

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

3) NM\_Live\_Weather\_Condition (dateWeather (PK), temperature, humidity, precipitation)

Sample Data

|  |  |  |  |
| --- | --- | --- | --- |
| dateWeather (PK) | temperature | humidity | precipitation |
| 03/14/2024 05:55 | 55 | 13 | 0.00 |
| 03/15/2024 06:09 | 50 | 15 | 1.22 |
| 03/16/2024 05:50 | 54 | 9 | 2.01 |
| 03/17/2024 05:59 | 48 | 12 | 0.00 |
| 03/18/2024 06:10 | 44 | 34 | 1.03 |
| 03/19/2024 05:58 | 44 | 5 | 0.00 |
| 03/20/2024 05:58 | 56 | 12 | 0.09 |
| 03/21/2024 05:52 | 67 | 9 | 0.00 |

Key: dateTime

FD1: dateTime -> temperature, humidity, precipitation, deliveryNumber(FK)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

4) IIB\_Delivery\_Schedule (deliveryNumber (PK), gpsTracker, deliveryDuration, licenseNumber(FK), vehicleIdentificationNumber(FK), dateWeather)(FK)

Sample data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| deliveryNumber (PK) | gpsTracker | deliveryDuration | licenseNumber (FK) | vehicleIdentificationNumber (FK) | dateWeather (FK) |
| 02234 | 07 | 1 | NM 294587-09 | 4Y1SL65848Z411439 | 03/14/2024 05:55 |
| 02235 | 06 | 3 | NM 965489-02 | 4M2EL65648T444468 | 03/15/2024 06:09 |
| 02245 | 13 | 2 | NM 145286-78 | 5H2RK789648T12446 | 03/16/2024 05:50 |
| 02344 | 11 | 1 | NM 064023-12 | 1T1QL12848N400032 | 03/17/2024 05:59 |
| 02349 | 87 | 2 | NM 770290-12 | 274NME3057THDF34 | 03/18/2024 06:10 |
| 02448 | 45 | 3 | NM 348206-97 | 283WUVH3489EUG6 | 03/19/2024 05:58 |
| 02449 | 23 | 1 | NM 082597-20 | 978UOI3850NMN593 | 03/20/2024 05:58 |
| 02350 | 67 | 4 | NM 503678-99 | 364HSN4086NRI3334 | 03/21/2024 05:52 |

Key: deliveryNumber

FD1: deliveryNumber -> gpsTracker, deliveryDuration

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

5) Order\_from\_STII\_to\_IIB (orderNumber(PK), iceCreamFlavors(PK), flavorQuantity, flavorPrice, neededByDate, deliveryNumber(FK))

Sample Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| orderNumber (PK) | iceCreamFlavors | flavorQuantity | flavorPrice | neededByDate | deliveryNumber (FK) |
| 00448 | Chocolate | 10 | 20 | 5/5/2024 | 02234 |
| 00449 | Vanilla | 10 | 20 | 5/5/2024 | 02235 |
| 00449 | Strawberry | 6 | 12 | 5/5/2024 | 02245 |
| 00451 | Rocky Road | 4 | 16 | 5/9/2024 | 02344 |
| 00452 | Strawberry | 2 | 4 | 5/9/2024 | 02349 |
| 00453 | Neapolitan | 4 | 16 | 5/10/2024 | 02448 |
| 00453 | Vanilla | 5 | 10 | 5/10/2024 | 02449 |
| 00455 | Chocolate | 5 | 10 | 5/10/2024 | 02350 |

Key: orderNumber, iceCreamFlavors

FD1: orderNumber, iceCreamFlavors -> flavorQuantity, flavorPrice, neededByDate, deliveryNumber(FK)

**FD2: iceCreamFlavors, flavorQuantity ->** flavorPrice

1NF: Meets the definition of a relation

2NF: Partial Key dependencies exist: orderNumber -> iceCreamFlavors and iceCreamFlavors -> flavorQuantity, flavorPrice

Solution: Split Order\_from\_STII\_to\_IIB into two new relations named order\_Data and iceCream\_Ordered:

order\_Data (orderNumber(PK), neededByDate, deliveryNumber(FK))

Sample Data

|  |  |  |
| --- | --- | --- |
| orderNumber (PK) | neededByDate | deliveryNumber (FK) |
| 00448 | 5/5/2024 | 02234 |
| 00449 | 5/5/2024 | 02235 |
| 00450 | 5/5/2024 | 02245 |
| 00451 | 5/9/2024 | 02344 |
| 00452 | 5/9/2024 | 02349 |
| 00453 | 5/10/2024 | 02448 |
| 00454 | 5/10/2024 | 02449 |
| 00455 | 5/10/2024 | 02350 |

Key: orderNumber

FD1: orderNumber -> neededByDate, deliveryNumber(FK)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

iceCream\_Ordered (iceCreamFlavors (PK), flavorQuantity, flavorPrice, orderNumber (FK))

Sample Data

|  |  |  |  |
| --- | --- | --- | --- |
| iceCreamFlavors (PK) | flavorQuantity | flavorPrice | orderNumber (FK) |
| Chocolate | 10 | 20 | 00448 |
| Vanilla | 10 | 20 | 00448 |
| Strawberry | 6 | 12 | 00450 |
| Rocky Road | 4 | 16 | 00450 |
| Mint Chocolate Chip | 2 | 8 | 00452 |
| Neapolitan | 4 | 16 | 00453 |
| Peanut Butter Chunk | 5 | 20 | 00454 |
| Blueberry Crumble | 5 | 20 | 00455 |

Key: iceCreamFlavors

FD1: iceCreamFlavors, flavorQuantity -> flavorPrice

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

6) STII\_Managers (STIImanagementNumber(PK), freezerNumber(FK), orderNumber(FK), shiftID(FK), employeeNumber(FK), deliveryNumber(FK))

Sample Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STIImanagementNumber (PK) | freezerNumber (FK) | orderNumber (FK) | shiftID (FK) | deliveryNumber(FK) |
| 0001 | 331 | 00448 | 3382 | 02234 |
| 0002 | 332 | 00449 | 3383 | 02235 |
| 0003 | 333 | 00450 | 3384 | 02245 |
| 0004 | 334 | 00451 | 3385 | 02344 |
| 0005 | 335 | 00452 | 3386 | 02349 |
| 0006 | 336 | 00453 | 3387 | 02448 |
| 0007 | 337 | 00454 | 3388 | 02449 |
| 0008 | 338 | 00455 | 3389 | 02350 |

Key: STIImanagementNumber

FD1: STIImanagementNumber -> freezerNumber(FK), orderNumber(FK), shiftID(FK), deliveryNumber(FK))

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

7) STII\_Managers\_ STII\_Employee(STIImanagementNumber(PK), employeeNumber(PK))

Sample Data

|  |  |
| --- | --- |
| STIImanagementNumber (PK) | employeeNumber (PK) |
| 0001 | 45478 |
| 0002 | 45433 |
| 0003 | 45479 |
| 0004 | 22012 |
| 0005 | 45476 |
| 0006 | 22013 |
| 0007 | 45432 |
| 0008 | 45477 |

Keys: STIImanagementNumber(PK), employeeNumber (PK)

FD1: STIImanagementNumber(PK), employeeNumber (PK)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

8) STII\_Staffing\_Schedule(shiftID(PK), shiftDate, shiftClockIn, ShiftClockout)

Sample Data

|  |  |  |  |
| --- | --- | --- | --- |
| shiftID (PK) | shiftDate | shiftClockIn | ShiftClockout |
| 3382 | 4/5/2024 | 12:01 | 16:06 |
| 3383 | 4/5/2024 | 16:05 | 19:01 |
| 3384 | 4/6/2024 | 12:00 | 16:01 |
| 3385 | 4/6/2024 | 16:00 | 19:02 |
| 3386 | 4/7/2024 | 12:06 | 16:05 |
| 3387 | 4/7/2024 | 16:01 | 19:00 |
| 3388 | 4/8/2024 | 12:02 | 16:09 |
| 3389 | 4/8/2024 | 16:03 | 19:10 |

Key: shiftID

FD1: shiftID -> shiftDate, shiftClockIn, ShiftClockout

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

9) STII\_Employee(employeeNumber(PK), shiftID(FK))

Sample Data

|  |  |
| --- | --- |
| employeeNumber (PK) | shiftID(FK) |
| 45478 | 3382 |
| 45433 | 3383 |
| 45479 | 3384 |
| 22012 | 3385 |
| 45476 | 3386 |
| 22013 | 3387 |
| 45432 | 3388 |
| 45477 | 3389 |

Key: employeeNumber

FD1: employeeNumber -> shiftID(FK)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

10) STII\_Freezer(freezerNumber(PK), temperature, employeeNumber(FK))

Sample Data

|  |  |  |
| --- | --- | --- |
| freezerNumber (PK) | temperature | employeeNumber(FK) |
| 331 | -10.9 | 45478 |
| 332 | -11.4 | 45433 |
| 333 | -19.7 | 45479 |
| 334 | -11.5 | 22012 |
| 335 | -11.4 | 45476 |
| 336 | -11.2 | 22013 |
| 337 | -14.9 | 45432 |
| 338 | -12.8 | 45477 |

Key: freezerNumber

FD1: freezerNumber -> temperature, employeeNumber(FK)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependency exists

BCNF: All determinants are candidate keys

**Final Set of Relations**

1) **IIB\_TruckDriver** (licenseNumber(PK), shiftNumber, firstName, LastName)

2) **IIB\_Truck** (vehicleIdentificationNumber(PK), lastDrivenByLicenseNumber)

3) **NM\_Live\_Weather\_Condition** (dateWeather(PK), temperature, humidity, precipitation)

4) **IIB\_Delivery\_Schedule** (deliveryNumber (PK), gpsTracker, deliveryDuration)

5) **order\_Data** (orderNumber(PK), iceCreamFlavors (FK), neededByDate, deliveryNumber(FK))

6) **iceCream\_Ordered** (iceCreamFlavors (PK), flavorQuantity, flavorPrice)

7) **STII\_Managers** (STIImanagementNumber(PK), freezerNumber(FK), orderNumber(FK), shiftID(FK), employeeNumber(FK), deliveryNumber(FK))

8) **STII\_Managers\_STII\_Employee** (STIImanagementNumber(PK), employeeNumber(PK))

9) **STII\_Staffing\_Schedule** (shiftID(PK), shiftDate, shiftClockIn, ShiftClockout)

10) **STII\_Employee (**employeeNumber(PK), shiftID(FK))

11) **STII\_Freezer** (freezerNumber(PK), temperature, employeeNumber(FK))

**Commentary on Normalization:**

* Note that only one normalization was required, splitting Order\_from\_STII\_to\_IIB into order\_Data and iceCream\_Ordered.

**V. Structured Query Language (SQL) to Create the Schema**

Then we created a table in the database for each of the relations in the final set of relations. We wrote and organized the SQL code by table.

DDL – Creating Tables

CREATE TABLE IIB\_Truck (

vehicleIdentificationNumber char(30) NOT NULL,

lastDrivenByLicenseNumber char(20) NOT NULL,

CONSTRAINT PK\_Truck PRIMARY KEY (vehicleIdentificationNumber)

);

CREATE TABLE IIB\_TruckDriver (

licenseNumber char(20) NOT NULL,

shiftNumber char(15) NOT NULL,

firstName char(15) NOT NULL,

lastName char(20) NOT NULL,

CONSTRAINT PK\_Driver PRIMARY KEY(licenseNumber)

);

CREATE TABLE NM\_Live\_Weather\_Condition (

dateWeather dateTime NOT NULL,

temperature numeric NOT NULL,

humidity int NOT NULL,

precipitation numeric NOT NULL,

CONSTRAINT PK\_Weather PRIMARY KEY (dateWeather)

);

CREATE TABLE STII\_Staffing\_Schedule (

shiftID char(10) NOT NULL,

shiftDate DateTime NOT NULL,

shiftClockIn DateTime NOT NULL,

shiftClockOut DateTime NOT NULL,

CONSTRAINT PK\_shiftID PRIMARY KEY(shiftID)

);

CREATE TABLE STII\_Employee (

employeeNumber char(10) NOT NULL,

shiftID char(10) NOT NULL,

CONSTRAINT PK\_STII\_Empl PRIMARY KEY (employeeNumber),

CONSTRAINT FK\_STII\_Empl\_Sched FOREIGN KEY (shiftID)

REFERENCES STII\_Staffing\_Schedule(shiftID)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE STII\_Freezer (

freezerNumber char(10) NOT NULL,

temperature numeric NOT NULL,

employeeNumber char(10) NOT NULL,

CONSTRAINT PK\_Freezer PRIMARY KEY (freezerNumber),

CONSTRAINT FK\_Empl\_Freezer FOREIGN KEY (employeeNumber)

REFERENCES STII\_Employee(employeeNumber)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IIB\_Delivery\_Schedule (

deliveryNumber char(15) NOT NULL,

gpsTracker char(10) NOT NULL,

deliveryDuration int NOT NULL,

licenseNumber char(20) NOT NULL,

vehicleIdentificationNumber char(30) NOT NULL,

dateWeather dateTime NOT NULL,

CONSTRAINT PK\_IIB\_Delivery PRIMARY KEY (deliveryNumber),

CONSTRAINT FK\_Driver\_Delivery FOREIGN KEY (licenseNumber)

REFERENCES IIB\_TruckDriver (licenseNumber)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Truck\_Delivery FOREIGN KEY (vehicleIdentificationNumber)

REFERENCES IIB\_Truck (vehicleIdentificationNumber)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Weather\_Delivery FOREIGN KEY (dateWeather)

REFERENCES NM\_Live\_Weather\_Condition (dateWeather)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE order\_data (

orderNumber char(10) NOT NULL,

neededByDate dateTime NOT NULL,

deliveryNumber char(15) NOT NULL,

CONSTRAINT PK\_Order PRIMARY KEY (orderNumber),

CONSTRAINT FK\_OrderDel FOREIGN KEY (deliveryNumber)

REFERENCES IIB\_Delivery\_Schedule(deliveryNumber)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE iceCream\_Ordered (

iceCreamFlavors char(25) NOT NULL,

flavorQuantity int NOT NULL,

flavorPrice currency NOT NULL,

orderNumber char(10) NOT NULL,

CONSTRAINT PK\_iceCream PRIMARY KEY (iceCreamFlavors),

CONSTRAINT FK\_iceCream FOREIGN KEY (orderNumber)

REFERENCES order\_data (orderNumber)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE STII\_Managers (

STIImanagementnumber char(10) NOT NULL,

freezerNumber char(10) NOT NULL,

orderNumber char(10) NOT NULL,

shiftID char(10) NOT NULL,

deliveryNumber char(15) NOT NULL,

CONSTRAINT PK\_Manager PRIMARY KEY (STIImanagementnumber),

CONSTRAINT FK\_Manager\_Freezer FOREIGN KEY (freezerNumber)

REFERENCES STII\_Freezer(freezerNumber)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Manager\_Order FOREIGN KEY (orderNumber)

REFERENCES order\_data(orderNumber)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Manager\_Staffing FOREIGN KEY (shiftID)

REFERENCES STII\_Staffing\_Schedule(shiftID)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Manager\_Delivery FOREIGN KEY (deliveryNumber)

REFERENCES IIB\_Delivery\_Schedule (deliveryNumber)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE STII\_Intersect\_Manager\_Employee (

employeeNumber char(10) NOT NULL,

STIImanagementnumber char(10) NOT NULL,

CONSTRAINT PK\_Interesect PRIMARY KEY (employeeNumber,STIImanagementnumber),

CONSTRAINT FK\_Intersect FOREIGN KEY (employeeNumber)

REFERENCES STII\_Employee (employeeNumber)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Intersect2 FOREIGN KEY (STIImanagementnumber)

REFERENCES STII\_Managers (STIImanagementnumber)

ON DELETE CASCADE ON UPDATE CASCADE

);

**Commentary on DDL**

* Keys are named with precision to properly represent the attribute they are referring to such as PK\_shiftID and FK\_freezernumber
* All keys in every table will update when data is deleted and updated to ensure the most recent records are shown to keep a steady delivery schedule if incidents occur.
* An intersect table was constructed between STIIEMPLOYEE and STIIMANAGERS to best represent the relationship between the two entities having a many too many relationships.
* It was decided Date and Time in tables that had such values should be separate attributes to ensure records are in greatest detail possible. Example, needing both shift clock in and clock out time in table STIISTAFFINGSCHDULE.

**VI. DML – Inserting Values into Tables**

INSERT INTO IIB\_TruckDriver VALUES ('NM 294587-09', '45879', 'William', 'Erickson');

INSERT INTO IIB\_TruckDriver VALUES ('NM 965489-02', '45880', 'Kent', 'Davidson');

INSERT INTO IIB\_TruckDriver VALUES ('NM 145286-78', '45881', 'Amy', 'Brookheimer');

INSERT INTO IIB\_TruckDriver VALUES ('NM 064023-12', '45882', 'Marjorie', 'Palmiotti');

INSERT INTO IIB\_TruckDriver VALUES ('NM 770290-12', '45883', 'Jonah', 'Ryan');

INSERT INTO IIB\_TruckDriver VALUES ('NM 348206-97', '45884', 'Catherine', 'Meyer');

INSERT INTO IIB\_TruckDriver VALUES ('NM 082597-20', '45885', 'Roger', 'Furlong');

INSERT INTO IIB\_TruckDriver VALUES ('NM 503678-99', '45886', 'Susan', 'Wilson');

INSERT INTO IIB\_Truck VALUES ('4Y1SL65848Z411439', 'NM 294587-09');

INSERT INTO IIB\_Truck VALUES ('4M2EL65648T444468', 'NM 065489-02');

INSERT INTO IIB\_Truck VALUES ('5H2RK789648T12446', 'NM 145286-78');

INSERT INTO IIB\_Truck VALUES ('1T1QL12848N400032', 'NM 064023-12');

INSERT INTO IIB\_Truck VALUES ('274NME3057THDF34', 'NM 770290-12');

INSERT INTO IIB\_Truck VALUES ('283WUVH3489EUG6', 'NM 348206-97');

INSERT INTO IIB\_Truck VALUES ('978UOI3850NMN593', 'NM 082597-20');

INSERT INTO IIB\_Truck VALUES ('364HSN4086NRI3334', 'NM 503678-99');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/14/2024 05:55', '55', '13', '0.00');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/15/2024 06:09', '50', '15', '1.22');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/16/2024 05:50', '54', '9', '2.01');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/17/2024 05:59', '48', '12', '0.00');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/18/2024 06:10', '44', '34', '1.03');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/19/2024 05:58', '44', '5', '0.00');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/20/2024 05:58', '56', '12', '0.09');

INSERT INTO NM\_Live\_Weather\_Condition VALUES ('03/21/2024 05:52', '67', '9', '0.00');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3382', '4/5/2024', '12:01', '16:06');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3383', '4/5/2024', '16:05', '19:01');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3384', '4/5/2024', '12:00', '16:01');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3385', '4/5/2024', '16:00', '19:02');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3386', '4/5/2024', '12:06', '16:05');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3387', '4/5/2024', '16:01', '19:00');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3388', '4/5/2024', '12:02', '16:09');

INSERT INTO STII\_Staffing\_Schedule VALUES ('3389', '4/5/2024', '16:03', '19:10');

INSERT INTO STII\_Employee VALUES ('45478', '3382');

INSERT INTO STII\_Employee VALUES ('45433', '3383');

INSERT INTO STII\_Employee VALUES ('45479', '3384');

INSERT INTO STII\_Employee VALUES ('22012', '3385');

INSERT INTO STII\_Employee VALUES ('45476', '3386');

INSERT INTO STII\_Employee VALUES ('22013', '3387');

INSERT INTO STII\_Employee VALUES ('45432', '3388');

INSERT INTO STII\_Employee VALUES ('45477', '3389');

INSERT INTO STII\_Freezer VALUES ('331', '-10.9', '45478');

INSERT INTO STII\_Freezer VALUES ('332', '-11.4', '45433');

INSERT INTO STII\_Freezer VALUES ('333', '-19.7', '45479');

INSERT INTO STII\_Freezer VALUES ('334', '-11.5', '22012');

INSERT INTO STII\_Freezer VALUES ('335', '-11.4', '45476');

INSERT INTO STII\_Freezer VALUES ('336', '-11.2', '22013');

INSERT INTO STII\_Freezer VALUES ('337', '-14.9', '45432');

INSERT INTO STII\_Freezer VALUES ('338', '-12.8', '45477');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02234', '07', '1', 'NM 294587-09', '4Y1SL65848Z411439', '03/14/2024 05:55');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02235', '06', '3', 'NM 965489-02', '4M2EL65648T444468 ', '03/15/2024 06:09');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02245', '13', '2', 'NM 145286-78', '5H2RK789648T12446', '03/16/2024 05:50 ');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02344', '11', '1', 'NM 064023-12', '1T1QL12848N400032', '03/17/2024 05:59');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02349', '87', '2', 'NM 770290-12', '274NME3057THDF34', '03/18/2024 06:10');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02448', '45', '3', 'NM 348206-97', '283WUVH3489EUG6', '03/19/2024 05:58');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02449', '23', '1', 'NM 082597-20', '978UOI3850NMN593 ', '03/20/2024 05:58');

INSERT INTO IIB\_Delivery\_Schedule VALUES ('02350', '67', '4', 'NM 503678-99', '364HSN4086NRI3334 ', '03/21/2024 05:52');

INSERT INTO order\_data VALUES ('00448', '5/5/2024', '02234');

INSERT INTO order\_data VALUES ('00449', '5/5/2024', '02235');

INSERT INTO order\_data VALUES ('00450', '5/5/2024', '02245');

INSERT INTO order\_data VALUES ('00451', '5/9/2024', '02344');

INSERT INTO order\_data VALUES ('00452', '5/9/2024', '02349');

INSERT INTO order\_data VALUES ('00453', '5/10/2024', '02448');

INSERT INTO order\_data VALUES ('00454', '5/10/2024', '02449');

INSERT INTO order\_data VALUES ('00455', '5/10/2024', '02350');

INSERT INTO iceCream\_Ordered VALUES ('Chocolate', '10', '20', '00448');

INSERT INTO iceCream\_Ordered VALUES ('Vanilla', '10', '20', '00449');

INSERT INTO iceCream\_Ordered VALUES ('Strawberry', '6', '12', '00450');

INSERT INTO iceCream\_Ordered VALUES ('Rocky Road', '4', '16', '00451');

INSERT INTO iceCream\_Ordered VALUES ('Mint Chocolate Chip', '2', '8', '00452');

INSERT INTO iceCream\_Ordered VALUES ('Neapolitan', '4', '16', '00453');

INSERT INTO iceCream\_Ordered VALUES ('Peanut Butter Chunk', '5', '20', '00454');

INSERT INTO iceCream\_Ordered VALUES ('Bluberry Crumble', '5', '20', '00455');

INSERT INTO STII\_Managers VALUES ('0001', '331', '00448', '3382', '02234');

INSERT INTO STII\_Managers VALUES ('0002', '332', '00449', '3383', '02235');

INSERT INTO STII\_Managers VALUES ('0003', '333', '00450', '3384', '02245');

INSERT INTO STII\_Managers VALUES ('0004', '334', '00451', '3385', '02344');

INSERT INTO STII\_Managers VALUES ('0005', '335', '00452', '3386', '02349');

INSERT INTO STII\_Managers VALUES ('0006', '336', '00453', '3387', '02448');

INSERT INTO STII\_Managers VALUES ('0007', '337', '00454', '3388', '02449');

INSERT INTO STII\_Managers VALUES ('0008', '338', '00455', '3389', '02350');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45478', '0001');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45433', '0002');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45479', '0003');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('22012', '0004');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45476', '0005');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('22013', '0006');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45432', '0007');

INSERT INTO STII\_Intersect\_Manager\_Employee VALUES ('45477', '0008');

**Commentary on DML Data Samples**

* We provided at least 8 rows of data in every table created, to make sure we can properly test the database when querying.
* We populated the database with a preview of relevant data for our client.
* With our populated database, we can begin querying the database.

**VII. DML – Queries that answer Business Questions**

**Make a list of truck driver names:**

SELECT firstName, lastName

FROM IIB\_TruckDriver;

**Make a list of freezers and their temperatures:**

SELECT freezerNumber, temperature

FROM STII\_Freezer;

**Make a list of freezers that had temperatures less than or equal to -14 degrees:**

SELECT freezerNumber, temperature

FROM STII\_Freezer

WHERE temperature <=-14;

**Make a list of which employees stocked which freezer:**

SELECT STII\_Employee.employeeNumber , STII\_Freezer.freezerNumber

FROM STII\_Freezer, STII\_Employee

WHERE STII\_Freezer.employeeNumber = STII\_Employee.employeeNumber

**What manager managed the latest shift?**

SELECT STIImanagementNumber

FROM STII\_Managers

WHERE shiftID =

(SELECT MAX(shiftID)

FROM STII\_Staffing\_Schedule);

**What is the DL number of the last driver who drove the vehicle with VIN 4Y1SL65848Z411439?**

SELECT lastDrivenByLicenseNumber

FROM IIB\_Truck

WHERE vehicleIdentificationNumber = '4Y1SL65848Z411439';

**Which ice cream flavors were ordered in 5-7 quantities?**

SELECT COUNT(iceCreamFlavors) AS myCount

FROM iceCream\_Ordered

WHERE flavorQuantity BETWEEN 5 AND 7;

**List cream flavors and their orders numbers:**

SELECT iceCream\_Ordered.iceCreamFlavors, order\_Data.OrderNumber

FROM iceCream\_Ordered, order\_Data

WHERE iceCream\_Ordered.OrderNumber = order\_Data.OrderNumber

**List dates where the temp was above 50 degrees:**

SELECT dateWeather, temperature

FROM NM\_Live\_Weather\_Condition

WHERE temperature > 50;

**Order shiftClockIn from latest to earliest:**

SELECT shiftClockIn

FROM STII\_Staffing\_Schedule

ORDER BY shiftClockIn DESC;

**Count the number of times each deliveryDuration occurs and group them:**

SELECT deliveryDuration, COUNT(\*) AS deliveryCount

FROM IIB\_Delivery\_Schedule

GROUP BY deliveryDuration;

**List orders where the total quantity of ice cream flavors are greater than 5:**

SELECT iceCreamFlavors, flavorQuantity, flavorPrice, OrderNumber

FROM iceCream\_Ordered

WHERE OrderNumber IN (

SELECT OrderNumber

FROM iceCream\_Ordered

GROUP BY OrderNumber

HAVING SUM(flavorQuantity) > 5

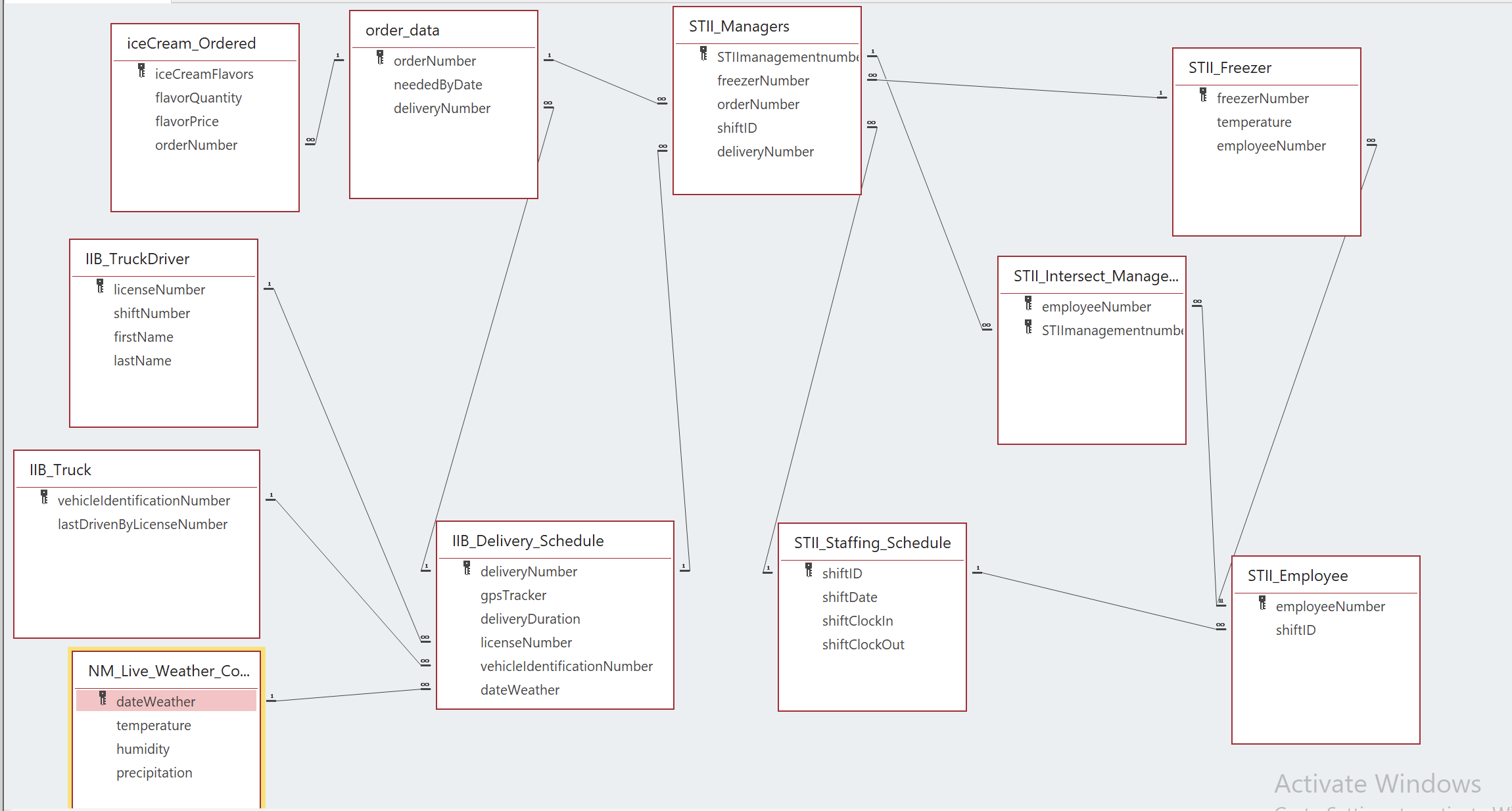
);

**Commentary on DML Practice Queries**

* Each table involved at least a single query, to ensure the entire database is fully operational without errors.
* Our database was able to handle statements including grouping, sub queries, joins, and conditional filtering successfully.
* Our queries answered relevant business questions that could help this DBMS be an effective tool for STTI.

**VII. Conclusion**

Our team has successfully designed and implemented a database schema requested by Scoop There It Is. We built the database to assist in managing ice cream deliveries, orders, staffing schedules, and managerial responsibilities. The database consists of several interconnected tables, including IIB\_Delivery\_Schedule, order\_data, iceCream\_Ordered, STII\_Managers, and STII\_Staffing\_Schedule, among others each normalized to 3NF to allow for fastest possible querying when using the database. The result of our database construction is pictured below, highlighting the relationships between tables.



Throughout the project, we utilized SQL queries to populate the database with relevant data and extract valuable insights. These queries cover a range of functionalities, including retrieving information about delivery schedules, orders, ice cream flavors, managerial responsibilities, and weather conditions as our client instructed us these are common queries they need performed. By employing subqueries, joins, grouping, aggregation, and conditional filtering, we've efficiently retrieved the desired information from the database when needed.

Overall, our group looked to create a database for Scoop There It Is that was efficient, reliable, yet highly accurate without duplicate information. We believe our database will help ensure our client has all the info they need to ensure a proper and timely delivery schedule from Ice Ice Baby. Our database is now fully constructed, tested, and fully operational to be used by our client.

**Meeting Timeline Log:**

|  |  |  |  |
| --- | --- | --- | --- |
| Date Met | Task Overlooked | Tasks Completed | Members Present |
| 3/5/24 | Data Base Idea, Basic Model | N/A | Wendy, Heather, Graham, Brandon |
| 3/15/24 | Start ER Model,  Start Normalization  Complete Basic Model | Data base Idea (Ice cream delivery)  Basic Model | Wendy, Heather, Graham, Brandon |
| 3/28/24 | Refine ER Model | ER Model | Heather, Wendy |
| 4/15/24 | Refine Normalization to start SQL | Normalization tables, sentences, description | Wendy, Heather, Graham, Brandon |
| 4/22/24 | Start SQL Code | Normalization and commentary of normalization | Brandon |
| 5/5/24 | Finish SQL Code  Write Up for sections 1-4 of project | SQL DDL, Sections 1-4 added commentary | Brandon, Wendy |
| 5/4/24 | Start Making Data Base, Add Insert statements and DML | SQL DDL, Ready for Data Base application | Graham |
| 5/8/24 | Finish Data Base, finish insert statements  Add commentary and write up to remaining sections of project | Data Base Complete, Relationships screenshot taken  Commentary added to rest of sections | Wendy, Heather, Graham |
| 5/14/24 | Finish DML and overlook Data base Model, Finish Commentary | DML, Commentary Added DDL and Insert statements  Log Sheet Finished | Wendy, Heather, Graham, Brandon |
| 5/16/24 | Final Overlook of Project | Project Complete | Wendy, Heather, Graham, Brandon |

Individual Tasks Completed:

1. Wendy finalized the Business Scenario.

2. Heather completed the ER Model using UML Notation.

3. Heather finished the Conversion to Relational Model.

4. Wendy finalized the Normalization task and sent the models to the Professor for Review.

5. Brandon completed the Structured Query Language (SQL) to Create the Schema.

6. Wendy reviewed part V and added language to the report as much as possible.

7. Graham added the SQL code to Access to create the Database.

8. Graham queried the INSERT INTO DML code to add data to the tables.

9. Brandon added/refined commentary and finalized conclusion.

Time Spent: Approximately 2 hours on each task.