Statement and Confirmation of Own Work

A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.

Student Declaration

I confirm the following details:

Candidate Name:	SHIHAB MIRZA
Candidate ID Number:	P00190603
Qualification:	NCC L5DC
Unit:	DDD (Database's design and development) (20 Credits)
Centre:	ZCAS University

I have read and understood both NCC Education's *Academic Misconduct Policy* and the *Referencing and Bibliographies* document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.

I confirm that this is my own work and that I have not colluded or plagiarised any part of it.

Candidate Signature:	
Date:	12/10/2022

TASK (1) (Business Description)-

Zambia Commodity Exchange (ZAMACE) is a commodity trading platform which involves an ecosystem of players that partake in it.

It enables the trade of many of the commodities that are grown and harvested in Zambia, like maize, soy bean, rice etc.
The players of this ecosystem involve, brokers, traders, farmers and warehouse operators.
A warehouse operator is simply a company that stores commodities under the provision of ZAMACE. The warehouse operator can have many storage sites (Warehouses) situated at various locations across the country. The storage sites have a certain storage capacity for storing commodities.
Farmers deposit their harvest by the warehouse operators through brokers in exchange for money or a receipt if they want to keep it as a security. Obviously, there is a fee for storing it. But this process is not only limited for farmers, but any other people who want to store and secure their commodities. ZAMACE is also responsible for the act of grading the commodities present in a certain warehouse.

ZAMACE also keeps records about its “hot” warehouse locations. Where warehouses are located the most.

Each of the storage sites for all warehouse operators registered with ZAMCE are graded and tested by ZAMACE to check for the quality of their infrastructure and the suitability for storing commodities. Factors such as cleanliness levels, moisture levels, and so on, are measured by ZAMACE operatives.

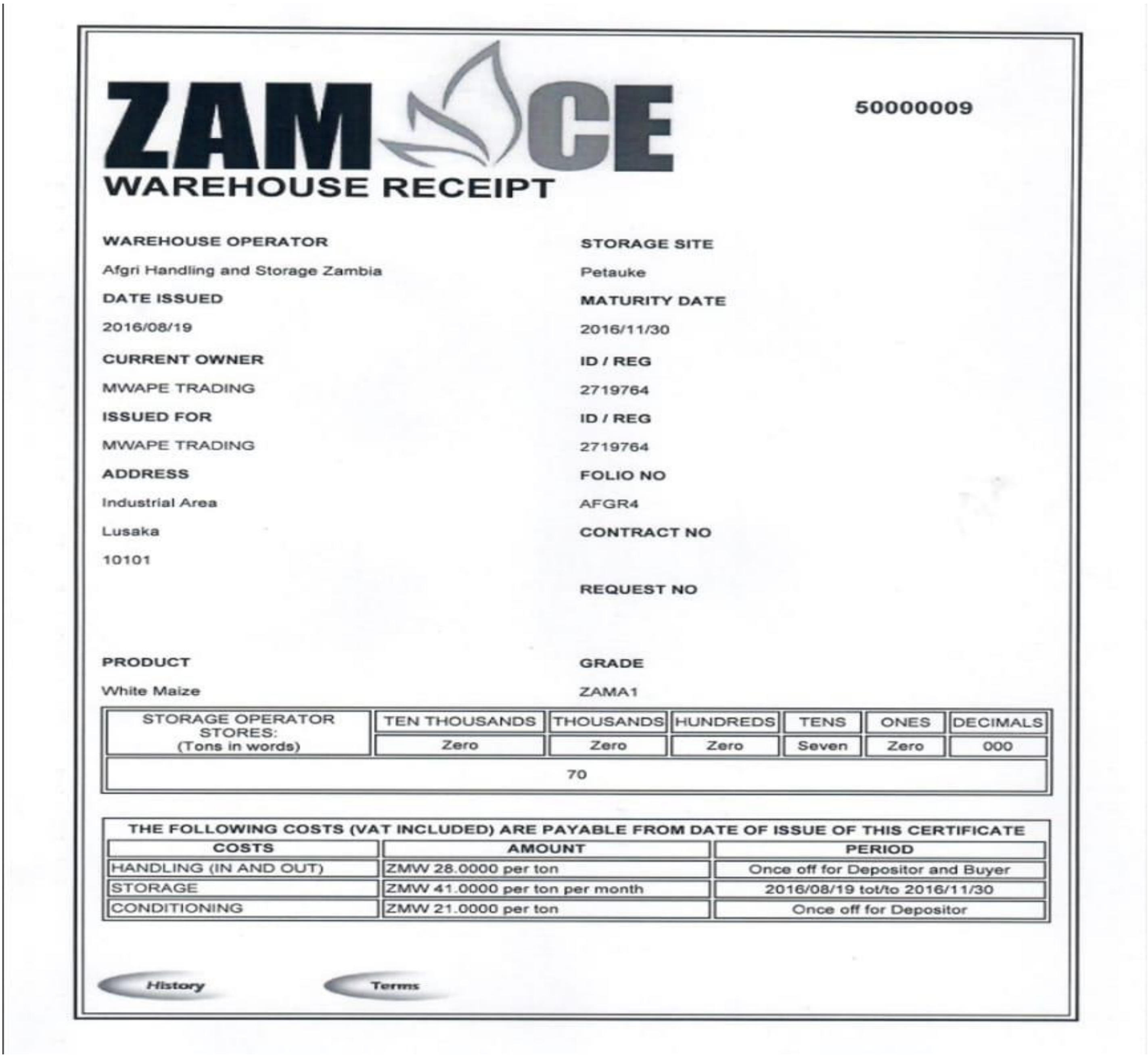
Brokers, are simply the middlemen or organizations that are present for allowing traders to execute trades. ZAMACE itself also has its own brokers.
ZAMACE takes a 1(one) % fee on each trade. And the brokers can have their own fees set. However, if one decides to choose a broker other than ZAMACE, he will not pay the 1% fee set by ZAMACE, rather the brokers will pay that. (Farmers can also be traders, but they are given a discount on storage fees if they are subsistence farmer hence ZAMACE keeps separate records for them). Commodity trades are only allowed in (metric tons). A broker cannot make more than 10000 trades on his account.

If a trader decides to buy a certain commodity, he will have the option to store it with a warehouse operator, for a certain fee per month. Until he finally decides to sell.
The constant daily buying and selling dictates the market price of each commodity.

The scope of this database project will critically consider data issues related to ZAMACE. Some of the entities are suggested below.

- Farmers
- Brokers
- Traders
- Warehouse Operators
- Commodities
- Storage sites
- Commodity grades.
- Locations
- Trades
- Storage site grades.

Below is a document from ZAMACE (a warehouse Receipt)



ZAMACE

50000009

WAREHOUSE RECEIPT

WAREHOUSE OPERATOR	**STORAGE SITE**
Afgri Handling and Storage Zambia	Petauke
DATE ISSUED	**MATURITY DATE**
2016/08/19	2016/11/30
CURRENT OWNER	**ID / REG**
MWAPE TRADING	2719764
ISSUED FOR	**ID / REG**
MWAPE TRADING	2719764
ADDRESS	**FOLIO NO**
Industrial Area	AFGR4
Lusaka	**CONTRACT NO**
10101	
	REQUEST NO
PRODUCT	**GRADE**
White Maize	ZAMA1

STORAGE OPERATOR STORES: (Tons in words)	TEN THOUSANDS	THOUSANDS	HUNDREDS	TENS	ONES	DECIMALS
	Zero	Zero	Zero	Seven	Zero	000
70						

THE FOLLOWING COSTS (VAT INCLUDED) ARE PAYABLE FROM DATE OF ISSUE OF THIS CERTIFICATE

COSTS	AMOUNT	PERIOD
HANDLING (IN AND OUT)	ZMW 28.0000 per ton	Once off for Depositor and Buyer
STORAGE	ZMW 41.0000 per ton per month	2016/08/19 tot/to 2016/11/30
CONDITIONING	ZMW 21.0000 per ton	Once off for Depositor

History

Terms

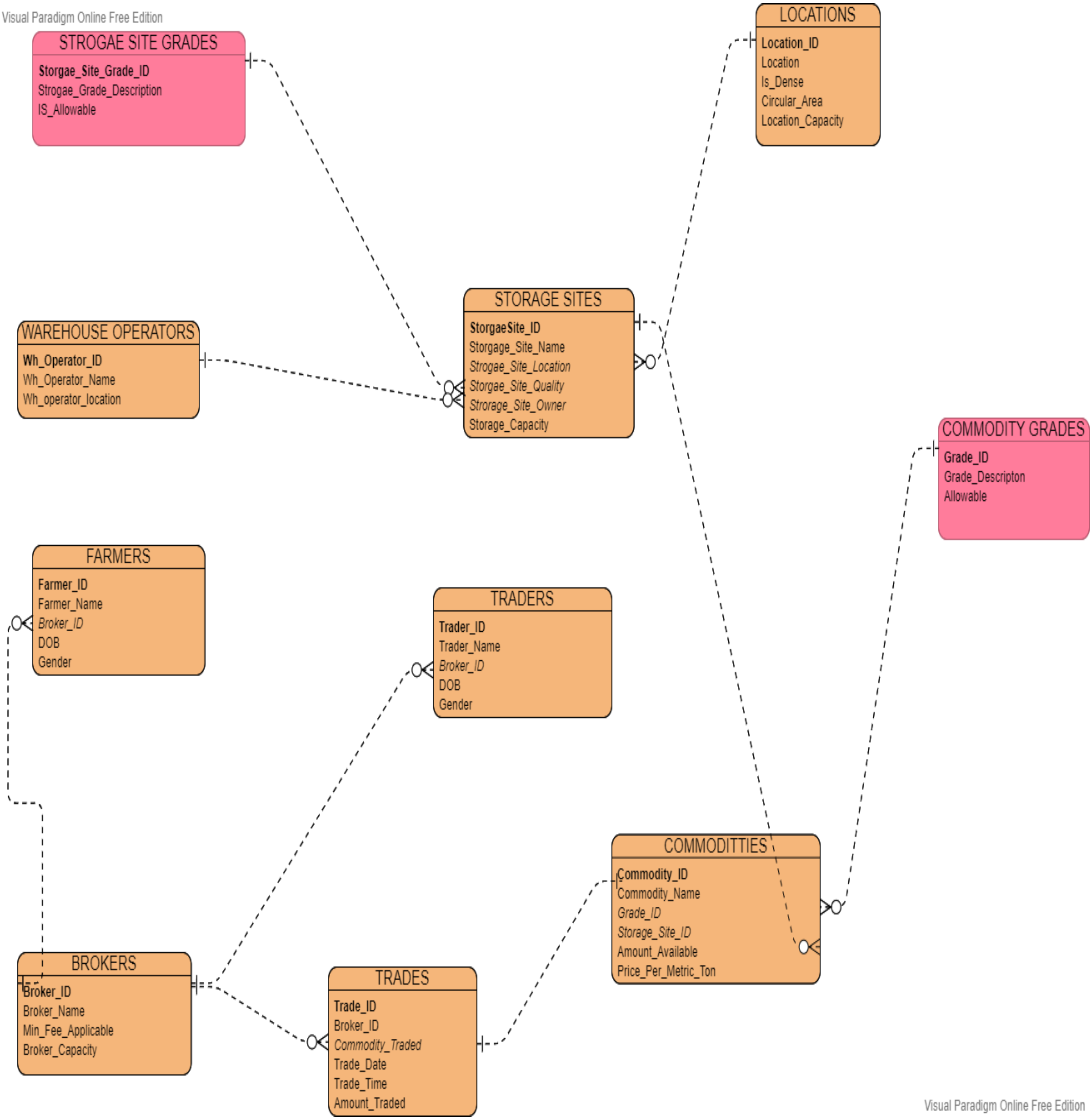
Note: It is important to know that not all of the attributes mentioned in this document might make it to the entity attributes. And also, that some new attributes might be added as well

ZAMACE has a complex set of mechanisms and data assets in its ecosystem which cannot be fully expressed through this project. Hence, the scope of this project will only include all attributes and pieces of data related to the entities mentioned above. All else will be ignored. For example, the trades that the brokers will place will be kept record of, but not what specific trader each trade belongs to. That is the duty of the brokers to handle.

TASK (2)-

ERD: (This image can be found in the assignment folder).

Hint: Primary keys are bold. Foreign keys are italic.



Some important constraints to note, just off the top:

A farmer or trader should be above the age of 18.

The valid values for gender are either male, female or other.

A Broker record on update cascade. On delete no action. In farmer and trader tables.

The valid values for the allowable fields in the pink tables are either True or False or Unknown.

<u>DATA DICTIONARY:</u>

All primary keys maintain entity integrity and foreign keys maintain relational and referential integrity…

TABLE	KEY	ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINTS	DESCRIPTION
FARMERS	Primary Key	Farmer ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each farmer
		Farmer Name	VARCHAR	40	Should be more than 1 char long	Name
		DOB	DATE	_	Should be in YYYY//MM//DD format	DOB
		Gender	VARCHAR	10	Should be male, female or other.	Gender
	Foreign key	Broker ID	VARCHAR	20	Foreign key, on delete set null, on update cascade	Tells us which broker the farmer is associated with
TRADERS	Primary Key	Trader ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each Trader
		Trader Name	VARCHAR	40	Should be more than 1 char long	Name
		DOB	DATE		Should be in YYYY//MM//DD format	DOB
		Gender	VARCHAR	10	Should be male, female or other.	Gender
	Foreign key	Broker ID	VARCHAR	20	Foreign key, on delete set null, on update cascade	Tells us which broker the trader is associated with
BROKERS	Primary Key	Broker ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each broker
		Broker Name	VARCHAR	30	Should be more than 1 char long, not null	Broker name
		Min_fee_applicable	FLOAT or DOUBLE	8 bytes	Should be more than 1	The minimum fee (in %) to pay to ZAMACE on each trade.
		Broker Capacity	INT	_	shouldn’t be more than 10000	The maximum number of trades a broker can have

TRADES	Primary Key	Trade ID	VARCHAR	20	Primary Key, Not Null	Uniquely iidentifies each trade
	Foreign key	Broker ID	VARCHAR	20	Foreign key, on delete no action, on update cascade	Identifies which broker placed trade.
	Foreign key	Commodity Traded	VARCHAR	20	Foreign key, on delete no action, on update cascade	Identifies which commodity was traded.
		Trade date	DATE	_	not null, should be in dd//mm//YY format	Date of trade
		Trade time	TIME	_	not null, should be in HH//MM//SS format	Time of trade
		Amount traded	INT	_	not null. Should be more than 10, less than 500, but correct to 0 DP and an int	amount traded in metric tons
COMMODITIES	Primary Key	Commodity ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each commodity
		Commodity name	VARCHAR	30	Should be valid commodity name	name of commodity
	Foreign key	Grade ID	VARCHAR	1	Foreign key, on delete no action, on update cascade	The grade given to this commodity
	Foreign key	Storage site ID	VARCHAR	20	Foreign key, on delete no action, on update cascade	The warehouse in which this commodity is present
		Amount_Available	INT	_	Should be more than 10000 (mt)	The amount of commodity present in (mt)

		Price_Per_Metric_Ton	DOUBLE	_	not null, should be correct to at least 2 DP	The price per metric ton for that commodity
COMMODITY GRADES	Primary Key	Grade ID	VARCHAR	1	Primary Key, Not Null,	Uniquely identifies each type of grading
		Grade Description	TEXT	100	Should be less than 100 chars	description
		Allowable	BOOLEAN	_	Allows only true, false or unknown value	If the grade given is acceptable or not.
WAREHOUSE OPERATORS	Primary Key	WH_operator ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each warehouse operator company
		WH_operator name	VARCHAR	30	not null	operator name
		Wh_operator location	VARCHAR	60	not null	the location of main office
STROAGE SITES	Primary Key	Storage site ID	VARCHAR	20	Primary Key, Not Null,	Uniquely identifies each storage site
		Storage site name	VARCHAR	30	Should be small caps	The name of the storage site
	Foreign key	Location ID	VARCHAR	20	Foreign key, on delete no action, on update cascade	The location of the storage site
	Foreign key	Quality	VARCHAR	1	Foreign key, on delete no action, on update cascade	The quality of the storage site
	Foreign key	Owner	VARCHAR	20	Foreign key, on delete set null, on update cascade	The owner (wwarehouse operator) of the storage site

		Capacity	INT	_	Not null, should be more than 500	The capacity in metric tons
STORAGE SITE GRADES	Primary Key	Grade ID	INT	11	Primary Key, Not Null	Uniquely identifies each type of grading
		Description	TEXT	100	Can be null	description
		Allowable	BOOLEAN	_	Allows only true, false or unknown value	If the grade given is acceptable or not.
LOCATIONS	Primary Key	Location ID	VARCHAR	20	Primary Key, Not Null	Uniquely identifies each location.
		Location	VARCHAR	60	Should be valid location	The name of the location
		is Dense	BOOLEAN	_	Allows only true, false or unknown value, not null	Checks if area is dense by checking the amount of storage sites present
		Circular area	DOUBLE	_	Should be more than 50 km squared	The circular area of the location in km squared
		Location capacity	DOUBLE	_	not null	The maximum number of warehouses the area can have.

TASK 3 (Normalization)

The method used for organizing data in a database is known as normalization. This method includes creating of tables and establishing of relationships between them in accordance with the rules that are meant to both safeguard the data and make the database overall more adaptable by removing of redundant and inconsistent dependencies.

removing data that is redundant (storing the same data in multiple tables) and making sure that data dependencies make sense (only storing related data in a table) are two important purposes.

There are 3 mainly know stages of normalization:

- 1-NF (Removing of repeating groups.)
- 2-NF (Removing of partial (non-key) dependencies).
- 3-NF (Removing of transitive dependencies.

All my entities are normalized to 3NF because due to the absence of insertion, update, and deletion anomalies in the majority of 3NF tables, Third Normal Form (3NF) is considered adequate for normal relational database design. Furthermore, 3NF always guarantees lossless and functional dependency preservation.

How I Verified that my tables are well structured using normalization:

- In my database, each of my tables has a single (atomic) value in each of its columns.
- The values stored in a column of each table are of the same domain; every table has its own name to avoid confusion; the order in which the data is stored is irrelevant; and normalization gives each column in each table a unique name.
- Each record is uniquely identified in my tables by the primary key because, there is functional dependence A -> B, C
- Update, deletion and insertion anomalies do not occur.

What causes an update anomaly?

If we changed data in one tuple and made it inconsistent with the same data in another tuple, we might get an update anomaly. Normalization solved this problem because, the data is structured by the database designer so that there are no duplicates, if duplication occurs, that will become another entity on its own. Given the example below. Let's say the commodities table was structured like this.

Commodity ID	Commodity Name	Grade Code	Grade Description	Allowable	Storage Site ID	Price ($)	Amount (mt)
1	Maize	A	Good	TRUE	1	200	1000
2	Soy Bean	C	Bad	FALSE	1	100	10000
3	Rice	C	Bad	FALSE	2	250	200000

Say now we want to change the Allowable values to TRUE and the Description to "consumable" for each commodity that has a C grade, we will have to change for all the records, which is redundant and brings about update anomalies.

So, we make our life easier by splitting the grading of the commodity into a separate table like so and the change is only made once;

Grade ID	Description	Allowable
A	GOOD	TRUE
B	OK	TRUE
C	CONSUMABLE	TRUE
D	BAD	FALSE

And we can just introduce a foreign key field in the commodity table like so:

Commodity ID	Commodity Name	Storage Site ID	Price ($)	Amount (mt)	GRADE ID
1	Maize	1	200	1000	A
2	Soy Bean	1	100	10000	C
3	Rice	2	250	200000	C

FOREIGN KEY FIELD

We can also take the storage sites table and storage site grades table as an example for this. Though for storage sites we use INT and we consider the higher number the better From 1 to 10.

TASK (4) (Assessment of design)

Logical Design

A conceptual, abstract design is a logical design. You haven't talked about the specifics of the actual implementation yet; Choose only the kinds of information you need.

The organization of data into a series of logical relationships known as entities and attributes is one aspect of logical design. A means of representing a piece of information is an entity. An entity and a table frequently correspond in relational databases. An attribute is a feature of an entity that makes it distinctive. A column is mapped to an attribute in relational databases.

The logical design can be created with a pen and paper or with a design tool like Oracle Designer or Oracle Warehouse Builder.

To physical design

This is what needs to be done to put the physical design into action, assuming that the logical data model is finished. By transforming the logical data model into a physical implementation based on our DBMS, we must establish an initial physical data model. We must accomplish the following in order to successfully create a physical data model: Attributes must be transformed into columns, relationships into foreign keys, and domains into constraints and data types, entities must all be transformed into tables.

How the tables have been designed:

In order to avoid data redundancies and anomalies, the designed database has been normalized to the third normal form. As a result, I have created ten tables, each with its own name and characteristics. Additionally, it has special keys like the foreign and primary keys. While the foreign key is used to connect one table to another and adheres to the referential integrity rules, the primary key is in charge of uniquely identifying each row in the table.

Derived Data:

New data is derived when existing raw data are combined and processed to produce derived data. Observational, experimental, and simulation data can be used to create derived data, but not previously derived data. In my case, the location table has a field called location capacity, the value for this column is derived from calculating the following, circular-area / average storage site land area. This column tells us the maximum number of warehouses possible to fit it that location.

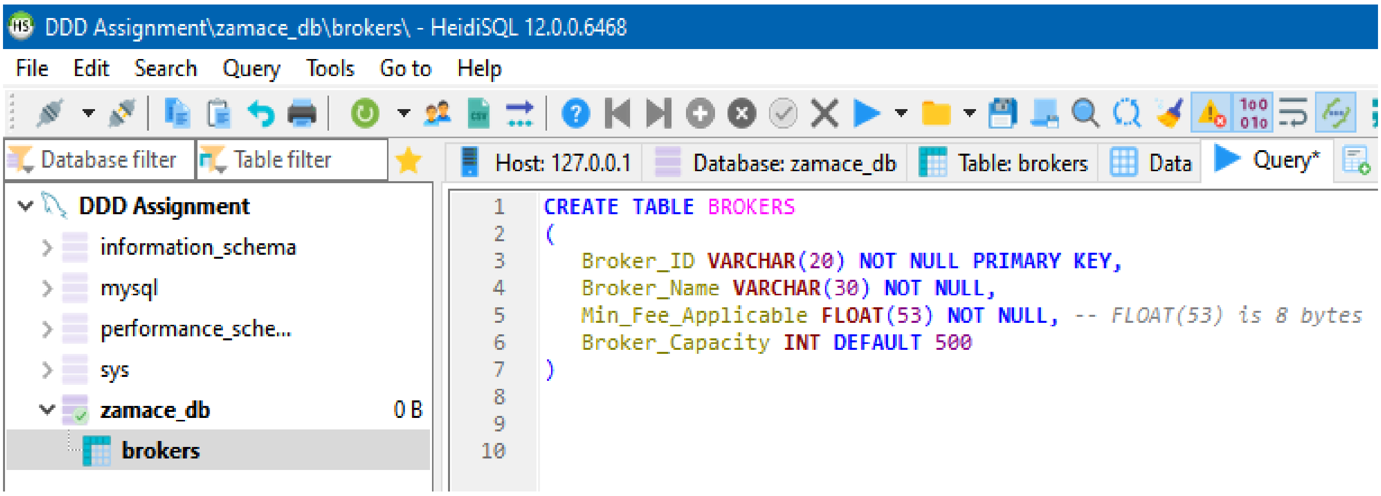
Denormalization:

Denormalization is a technique for improving performance in a normalized database. Denormalization is the process of trying to improve a database's read performance while sacrificing some write performance by grouping or adding redundant copies of data.

In my case, I will not be using any denormalization on any of my tables.

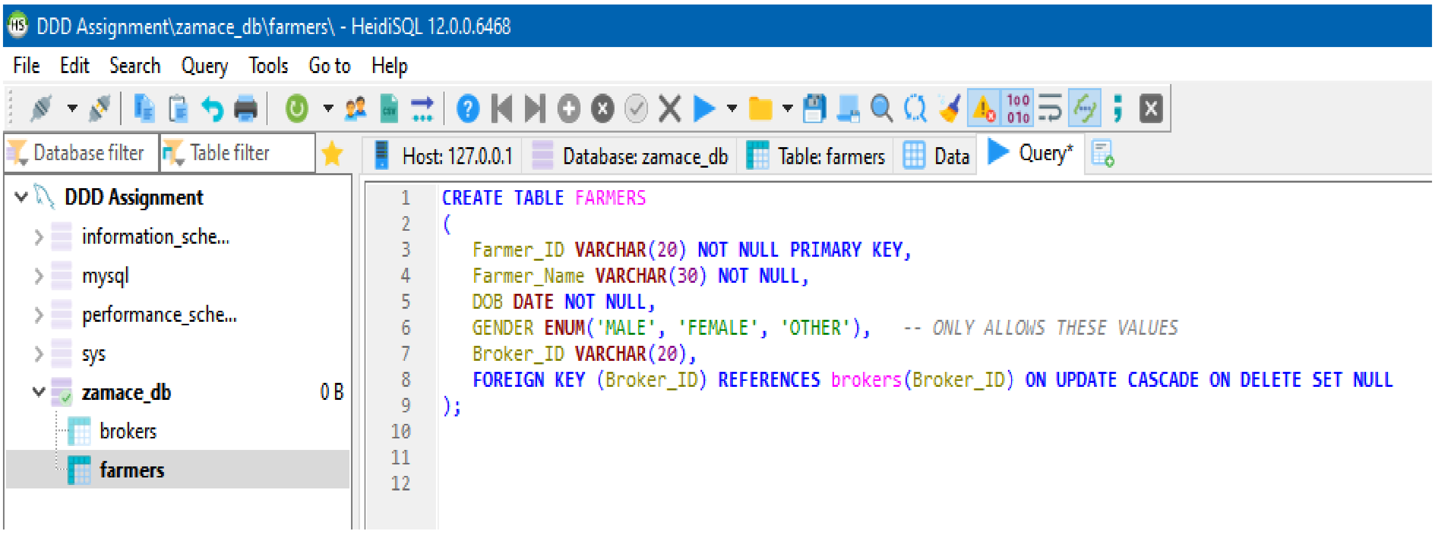
TASK (5) (Creating of tables)

BROKERS TABLE



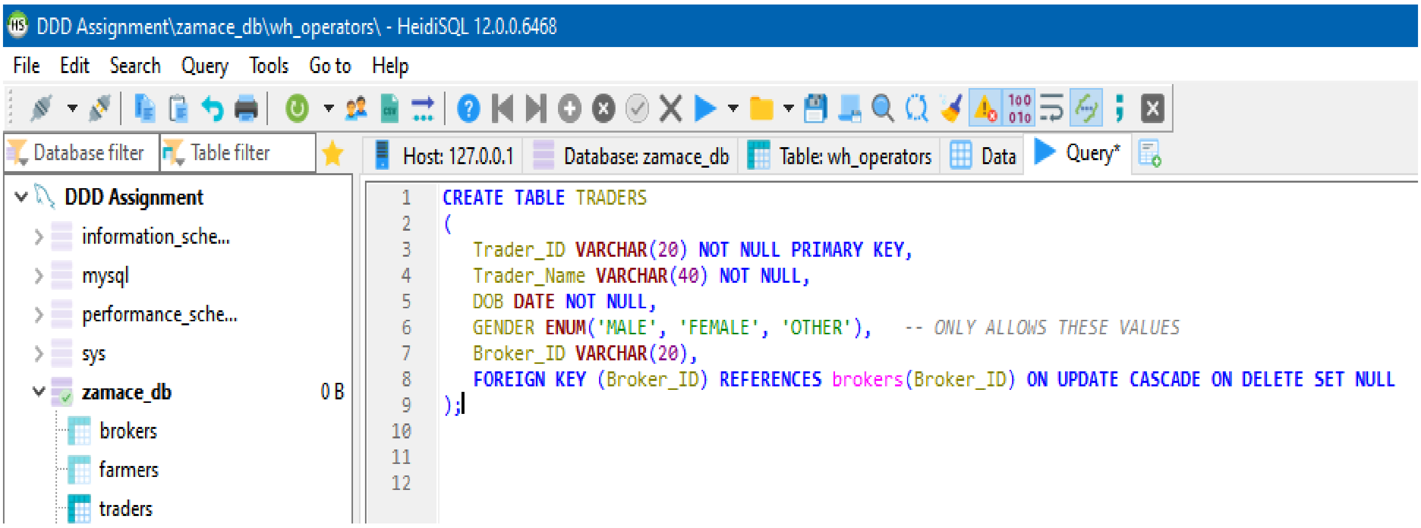
```sql
CREATE TABLE BROKERS
(
   Broker_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Broker_Name VARCHAR(30) NOT NULL,
   Min_Fee_Applicable FLOAT(53) NOT NULL, -- FLOAT(53) is 8 bytes
   Broker_Capacity INT DEFAULT 500
)
```

FARMERS TABLE



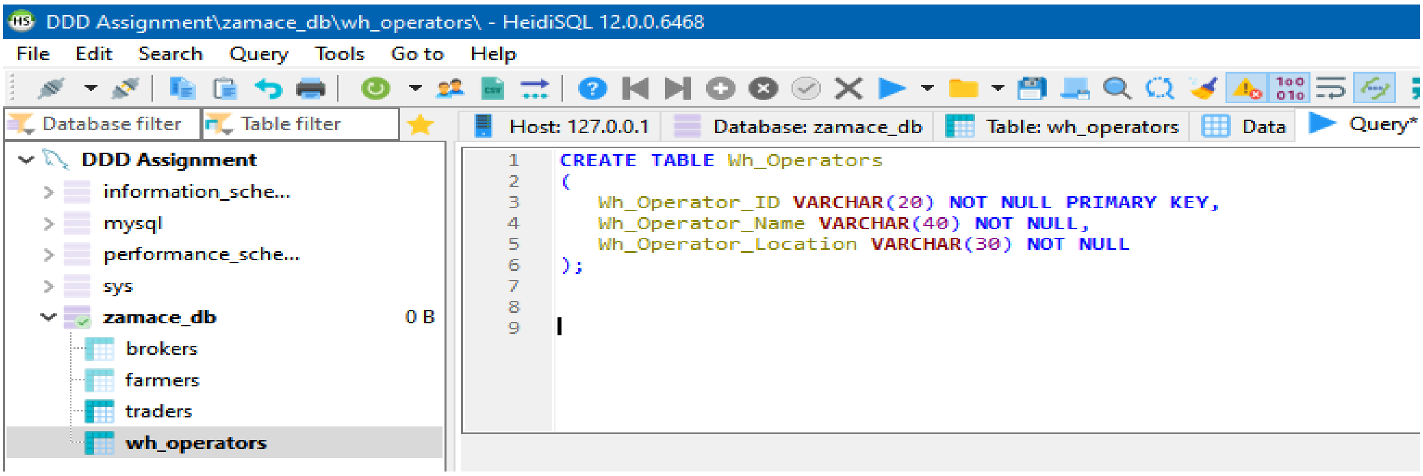
```sql
CREATE TABLE FARMERS
(
   Farmer_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Farmer_Name VARCHAR(30) NOT NULL,
   DOB DATE NOT NULL,
   GENDER ENUM('MALE', 'FEMALE', 'OTHER'),   -- ONLY ALLOWS THESE VALUES
   Broker_ID VARCHAR(20),
   FOREIGN KEY (Broker_ID) REFERENCES brokers(Broker_ID) ON UPDATE CASCADE ON DELETE SET NULL
);
```

TRADERS TABLE



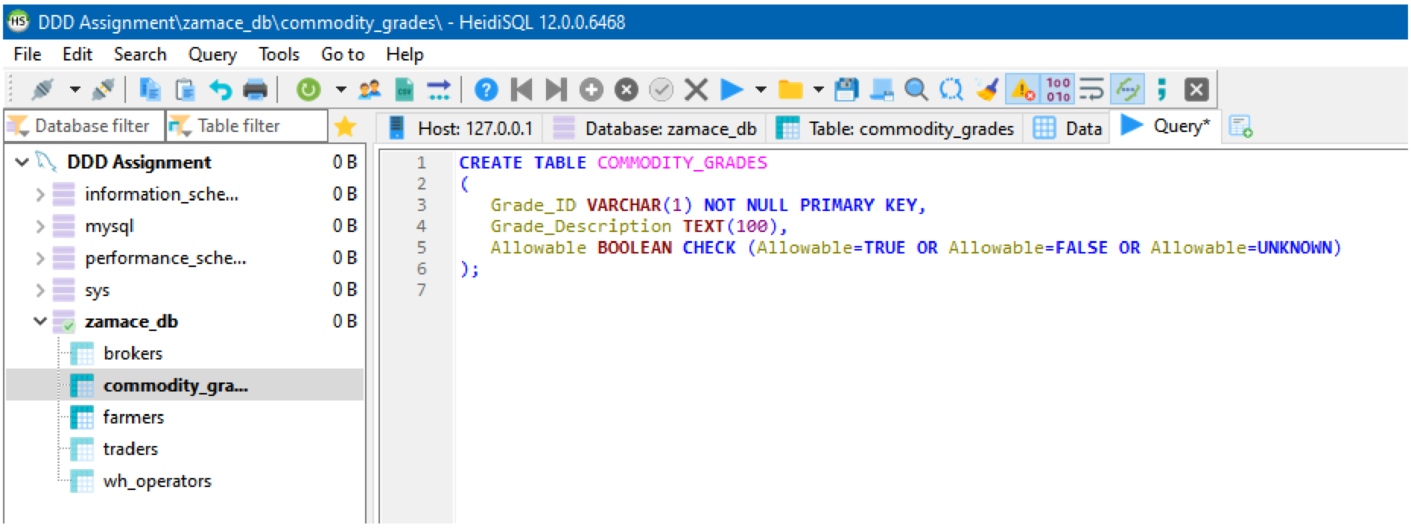
```sql
CREATE TABLE TRADERS
(
   Trader_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Trader_Name VARCHAR(40) NOT NULL,
   DOB DATE NOT NULL,
   GENDER ENUM('MALE', 'FEMALE', 'OTHER'),   -- ONLY ALLOWS THESE VALUES
   Broker_ID VARCHAR(20),
   FOREIGN KEY (Broker_ID) REFERENCES brokers(Broker_ID) ON UPDATE CASCADE ON DELETE SET NULL
);
```

WAREHOUSE OPERATORS TABLE



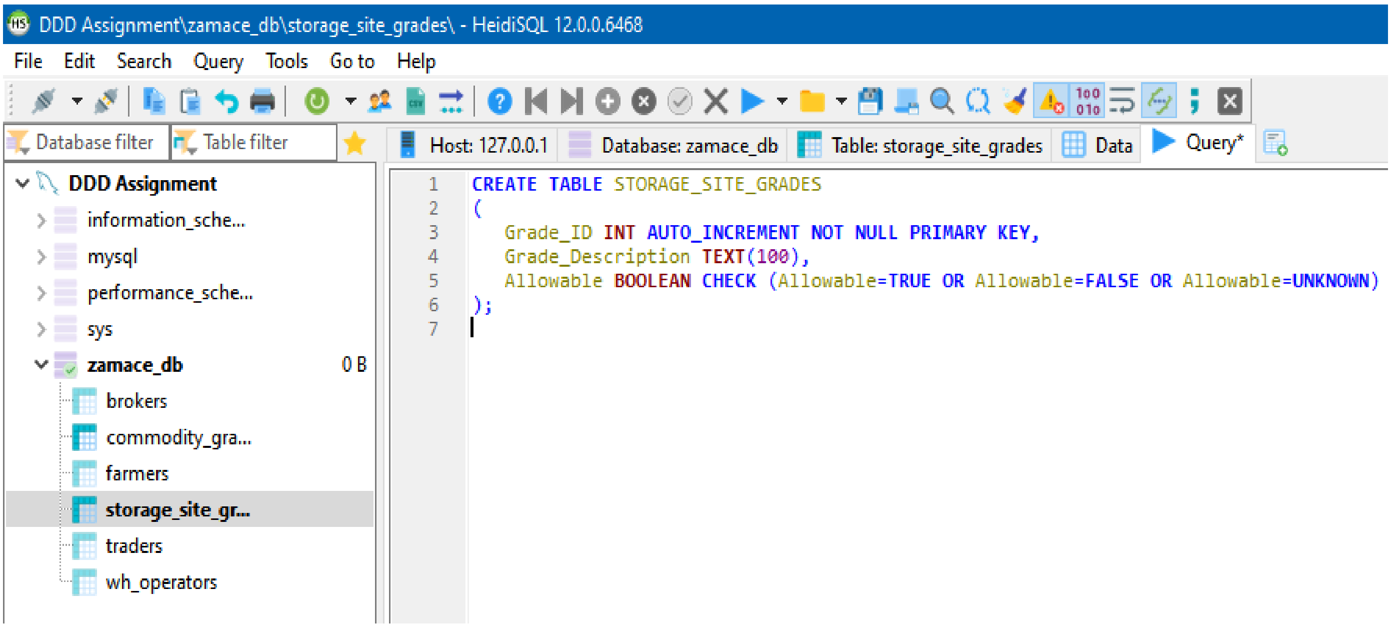
```
CREATE TABLE Wh_Operators
(
   Wh_Operator_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Wh_Operator_Name VARCHAR(40) NOT NULL,
   Wh_Operator_Location VARCHAR(30) NOT NULL
);
```

COMMODITY GRADES TABLE



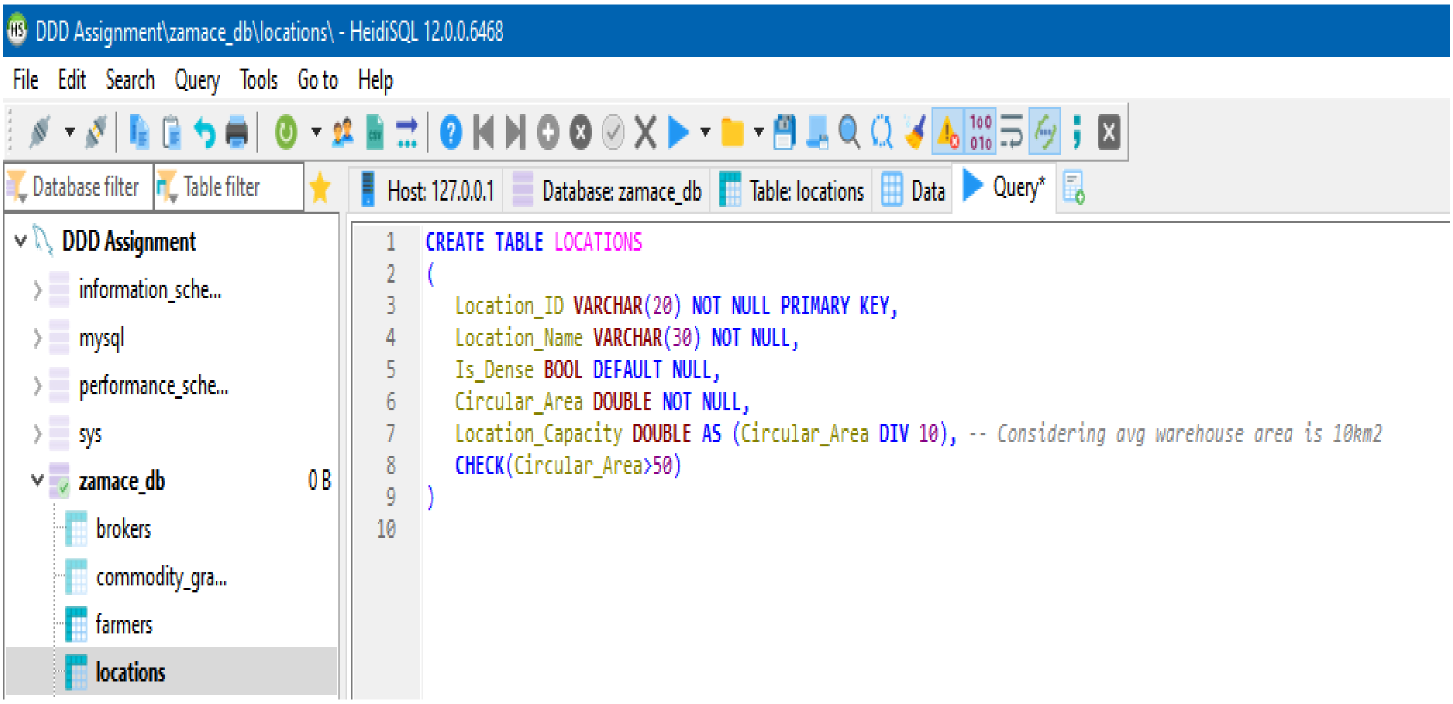
```
CREATE TABLE COMMODITY_GRADES
(
   Grade_ID VARCHAR(1) NOT NULL PRIMARY KEY,
   Grade_Description TEXT(100),
   Allowable BOOLEAN CHECK (Allowable=TRUE OR Allowable=FALSE OR Allowable=UNKNOWN)
);
```

STORAGE SITE GRADES TABLE



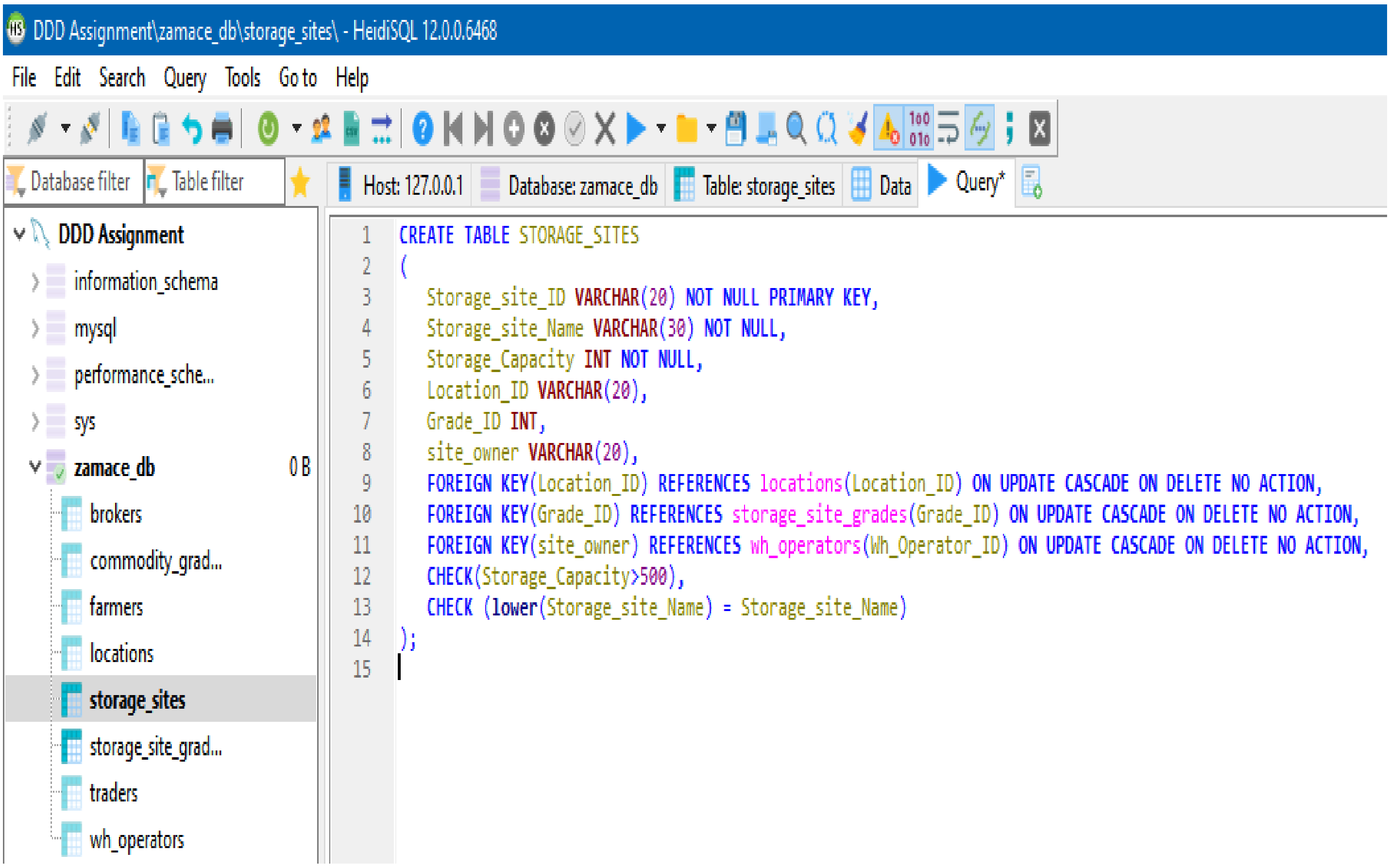
```
CREATE TABLE STORAGE_SITE_GRADES
(
   Grade_ID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
   Grade_Description TEXT(100),
   Allowable BOOLEAN CHECK (Allowable=TRUE OR Allowable=FALSE OR Allowable=UNKNOWN)
);
```

LOCATIONS TABLE



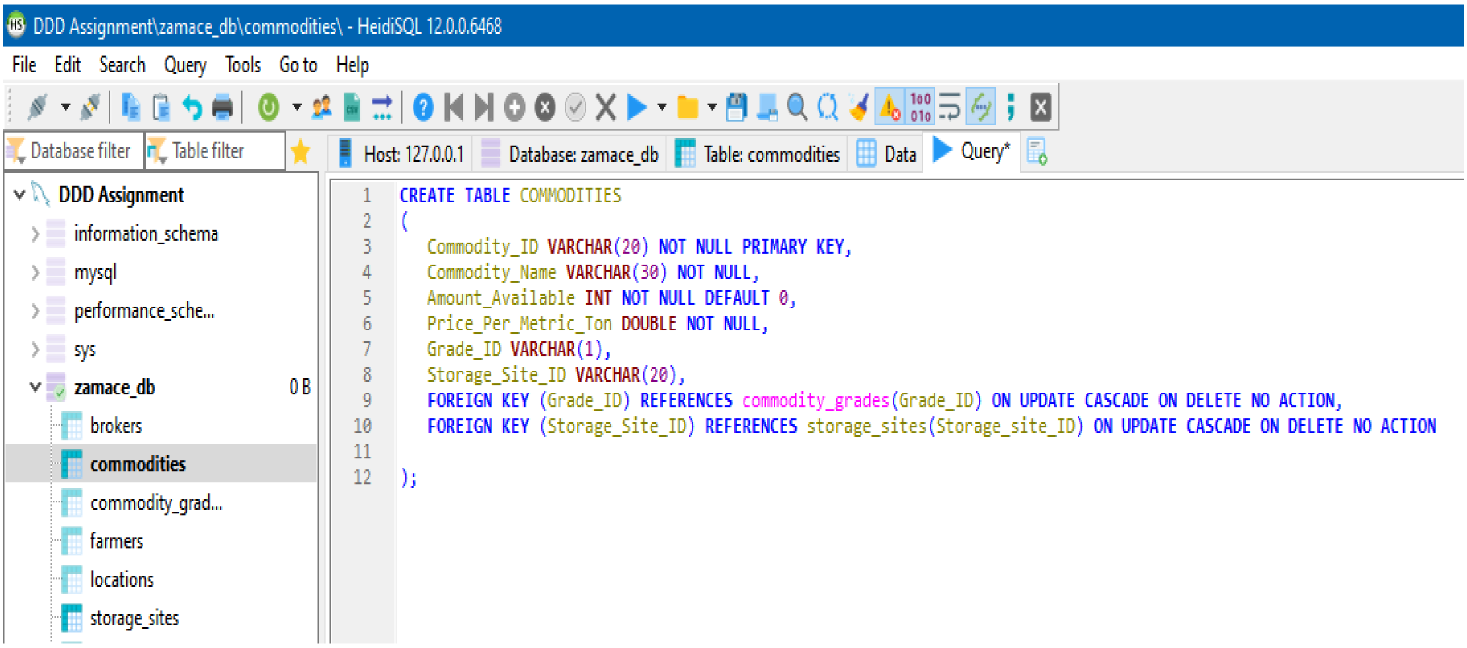
```
CREATE TABLE LOCATIONS
(
  Location_ID VARCHAR(20) NOT NULL PRIMARY KEY,
  Location_Name VARCHAR(30) NOT NULL,
  Is_Dense BOOL DEFAULT NULL,
  Circular_Area DOUBLE NOT NULL,
  Location_Capacity DOUBLE AS (Circular_Area DIV 10), -- Considering avg warehouse area is 10km2
  CHECK(Circular_Area>50)
)
```

STORAGE SITES TABLE



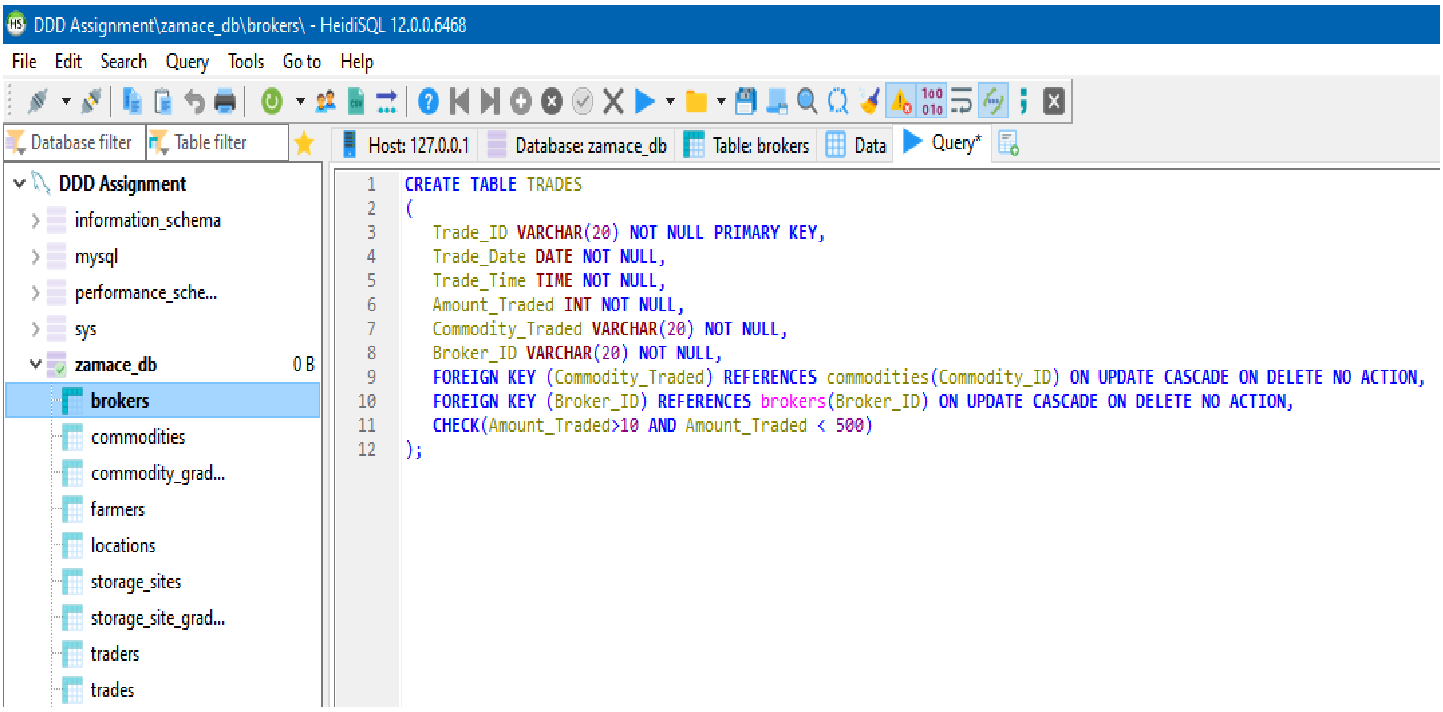
```
CREATE TABLE STORAGE_SITES
(
  Storage_site_ID VARCHAR(20) NOT NULL PRIMARY KEY,
  Storage_site_Name VARCHAR(30) NOT NULL,
  Storage_Capacity INT NOT NULL,
  Location_ID VARCHAR(20),
  Grade_ID INT,
  site_owner VARCHAR(20),
  FOREIGN KEY(Location_ID) REFERENCES locations(Location_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
  FOREIGN KEY(Grade_ID) REFERENCES storage_site_grades(Grade_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
  FOREIGN KEY(site_owner) REFERENCES wh_operators(Wh_Operator_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
  CHECK(Storage_Capacity>500),
  CHECK (lower(Storage_site_Name) = Storage_site_Name)
);
```

COMMODITIES TABLE



```
CREATE TABLE COMMODITIES
(
   Commodity_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Commodity_Name VARCHAR(30) NOT NULL,
   Amount_Available INT NOT NULL DEFAULT 0,
   Price_Per_Metric_Ton DOUBLE NOT NULL,
   Grade_ID VARCHAR(1),
   Storage_Site_ID VARCHAR(20),
   FOREIGN KEY (Grade_ID) REFERENCES commodity_grades(Grade_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
   FOREIGN KEY (Storage_Site_ID) REFERENCES storage_sites(Storage_site_ID) ON UPDATE CASCADE ON DELETE NO ACTION

);
```

TRADES TABLE



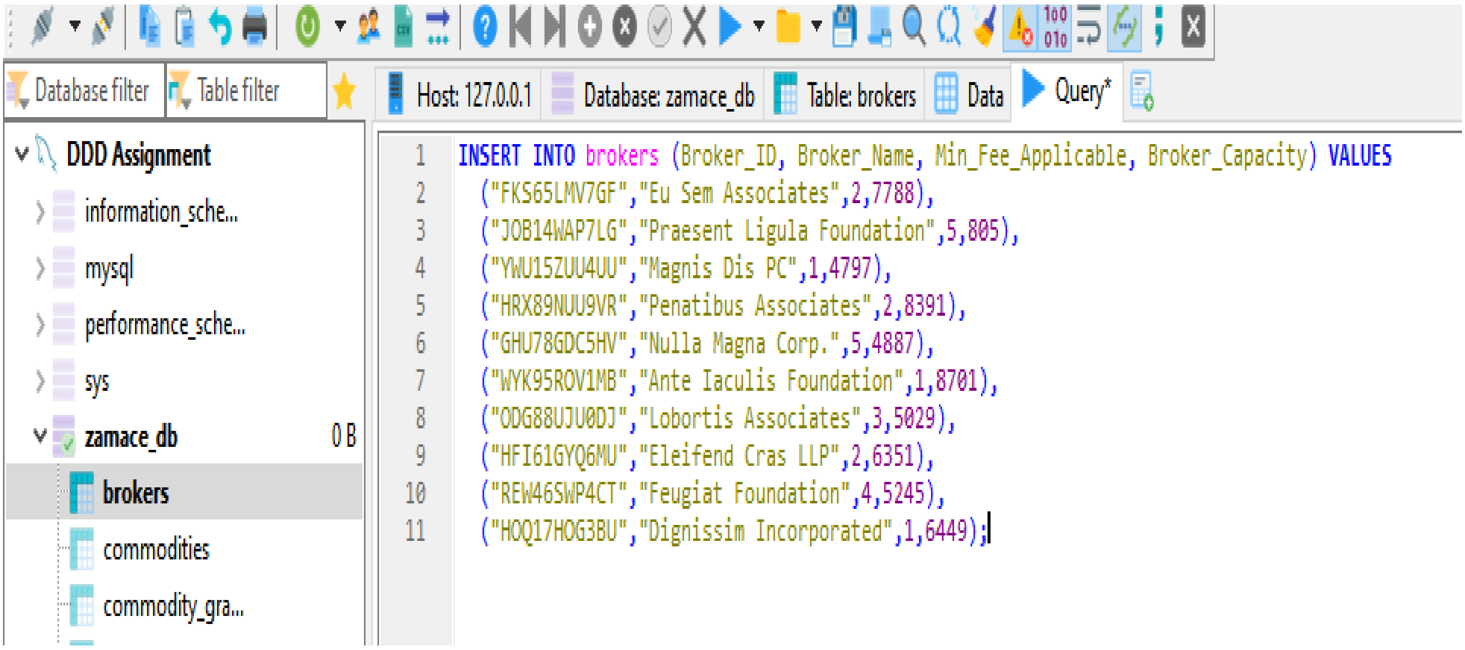
```
CREATE TABLE TRADES
(
   Trade_ID VARCHAR(20) NOT NULL PRIMARY KEY,
   Trade_Date DATE NOT NULL,
   Trade_Time TIME NOT NULL,
   Amount_Traded INT NOT NULL,
   Commodity_Traded VARCHAR(20) NOT NULL,
   Broker_ID VARCHAR(20) NOT NULL,
   FOREIGN KEY (Commodity_Traded) REFERENCES commodities(Commodity_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
   FOREIGN KEY (Broker_ID) REFERENCES brokers(Broker_ID) ON UPDATE CASCADE ON DELETE NO ACTION,
   CHECK(Amount_Traded>10 AND Amount_Traded < 500)
);
```

How I developed the scripts, order or running and errors I encountered:

I developed the scripts mainly with the help of the data dictionary and the Entity relational diagram. I checked for constraints, data types etc. and I translated them into SQL code, The order I which I created the tables was such that, I created tables that had no foreign keys, first, then I created the tables with foreign keys to avoid referencing errors. The errors I encountered were that some of the domain constraints were not supported in the SQL version I was using so I had to look for workarounds. Note that there might still be changes to be made in the future using ALTER TABLE.

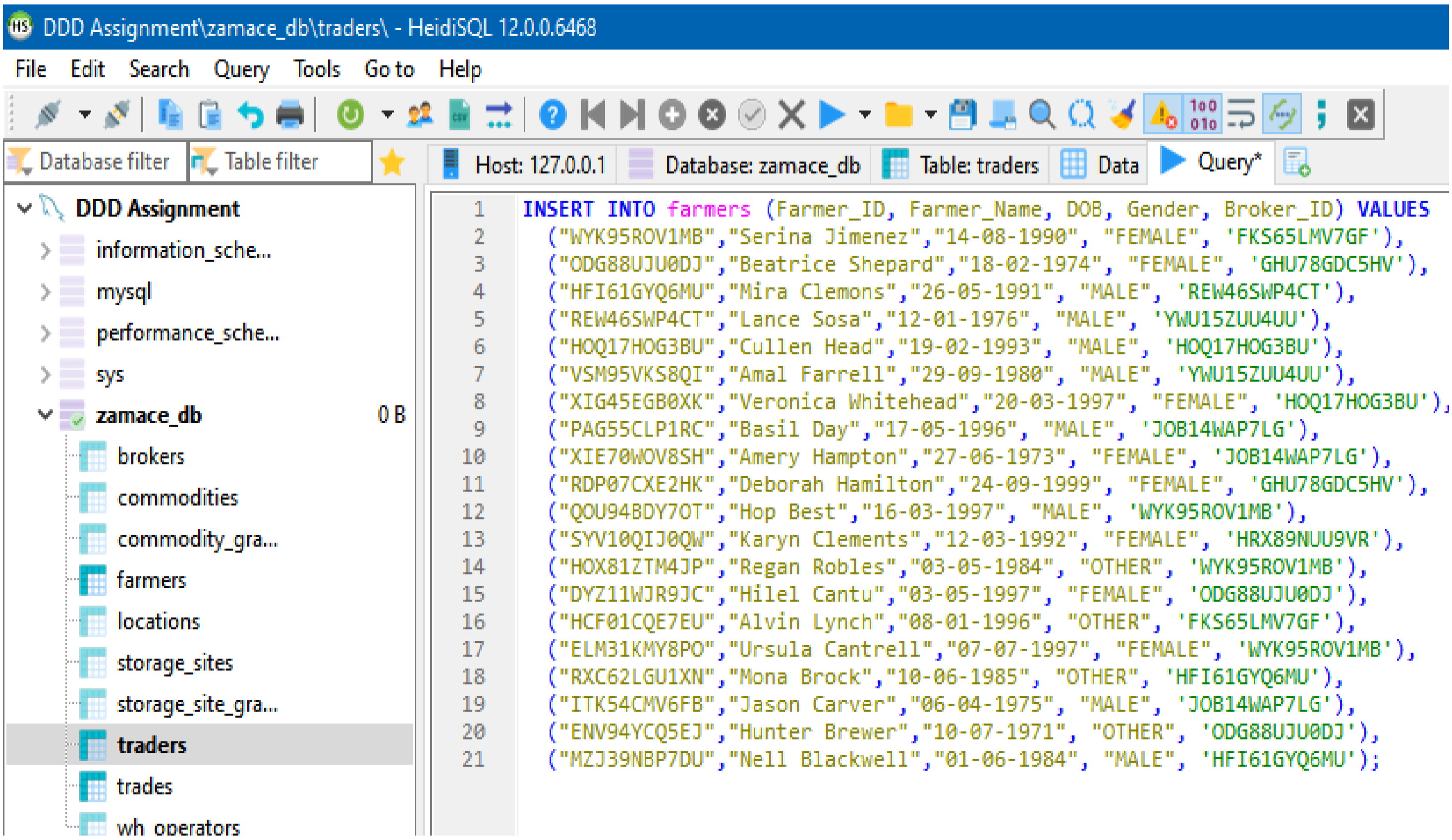
TASK (6) (Data Population)

BROKERS TABLE



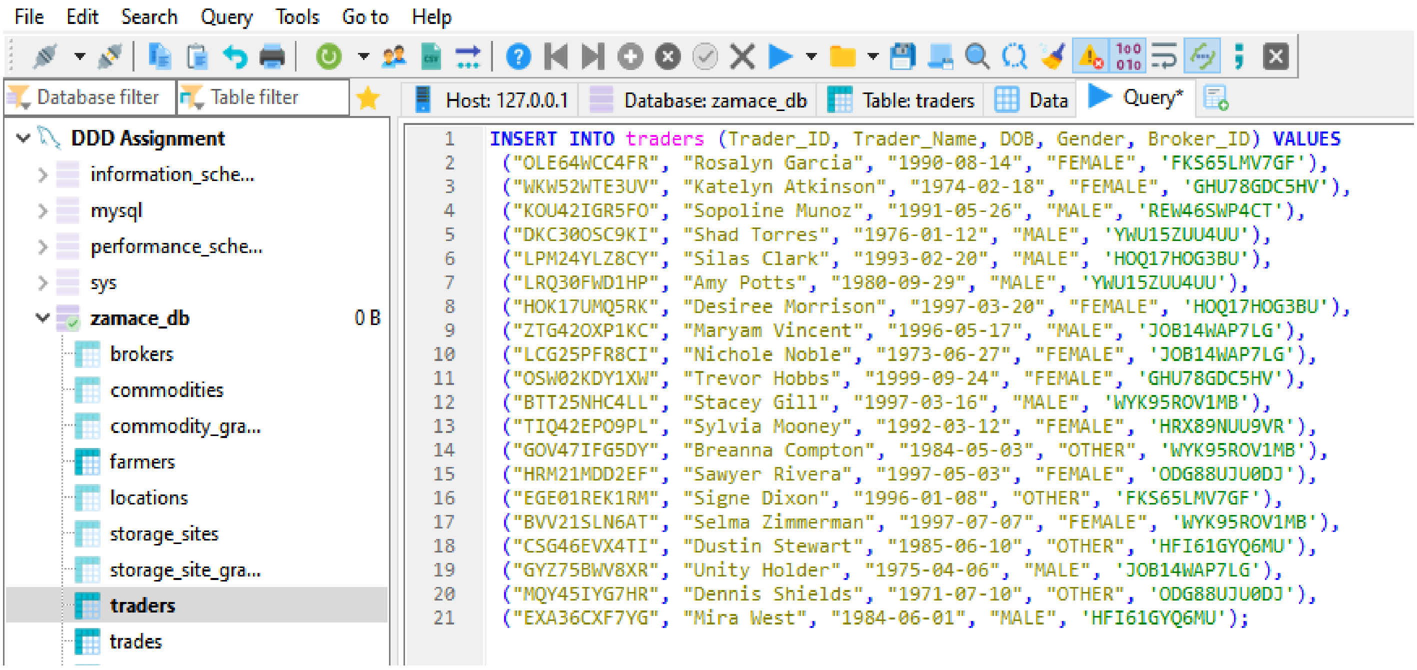
```sql
INSERT INTO brokers (Broker_ID, Broker_Name, Min_Fee_Applicable, Broker_Capacity) VALUES
  ("FKS65LMV7GF","Eu Sem Associates",2,7788),
  ("JOB14WAP7LG","Praesent Ligula Foundation",5,805),
  ("YWU15ZUU4UU","Magnis Dis PC",1,4797),
  ("HRX89NUU9VR","Penatibus Associates",2,8391),
  ("GHU78GDC5HV","Nulla Magna Corp.",5,4887),
  ("WYK95ROV1MB","Ante Iaculis Foundation",1,8701),
  ("ODG88UJU0DJ","Lobortis Associates",3,5029),
  ("HFI61GYQ6MU","Eleifend Cras LLP",2,6351),
  ("REW46SWP4CT","Feugiat Foundation",4,5245),
  ("HOQ17HOG3BU","Dignissim Incorporated",1,6449);
```

FARMERS TABLE



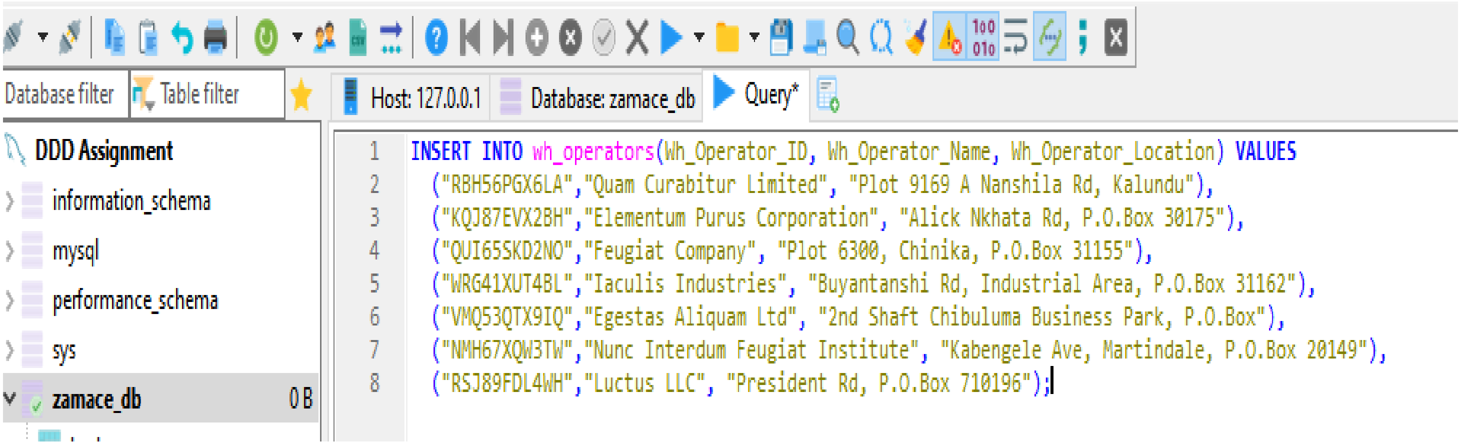
```sql
INSERT INTO farmers (Farmer_ID, Farmer_Name, DOB, Gender, Broker_ID) VALUES
  ("WYK95ROV1MB","Serina Jimenez","14-08-1990", "FEMALE", 'FKS65LMV7GF'),
  ("ODG88UJU0DJ","Beatrice Shepard","18-02-1974", "FEMALE", 'GHU78GDC5HV'),
  ("HFI61GYQ6MU","Mira Clemons","26-05-1991", "MALE", 'REW46SWP4CT'),
  ("REW46SWP4CT","Lance Sosa","12-01-1976", "MALE", 'YWU15ZUU4UU'),
  ("HOQ17HOG3BU","Cullen Head","19-02-1993", "MALE", 'HOQ17HOG3BU'),
  ("VSM95VKS8QI","Amal Farrell","29-09-1980", "MALE", 'YWU15ZUU4UU'),
  ("XIG45EGB0XK","Veronica Whitehead","20-03-1997", "FEMALE", 'HOQ17HOG3BU'),
  ("PAG55CLP1RC","Basil Day","17-05-1996", "MALE", 'JOB14WAP7LG'),
  ("XIE70WOV8SH","Amery Hampton","27-06-1973", "FEMALE", 'JOB14WAP7LG'),
  ("RDP07CXE2HK","Deborah Hamilton","24-09-1999", "FEMALE", 'GHU78GDC5HV'),
  ("QOU94BDY7OT","Hop Best","16-03-1997", "MALE", 'WYK95ROV1MB'),
  ("SYV10QIJ0QW","Karyn Clements","12-03-1992", "FEMALE", 'HRX89NUU9VR'),
  ("HOX81ZTM4JP","Regan Robles","03-05-1984", "OTHER", 'WYK95ROV1MB'),
  ("DYZ11WJR9JC","Hilel Cantu","03-05-1997", "FEMALE", 'ODG88UJU0DJ'),
  ("HCF01CQE7EU","Alvin Lynch","08-01-1996", "OTHER", 'FKS65LMV7GF'),
  ("ELM31KMY8PO","Ursula Cantrell","07-07-1997", "FEMALE", 'WYK95ROV1MB'),
  ("RXC62LGU1XN","Mona Brock","10-06-1985", "OTHER", 'HFI61GYQ6MU'),
  ("ITK54CMV6FB","Jason Carver","06-04-1975", "MALE", 'JOB14WAP7LG'),
  ("ENV94YCQ5EJ","Hunter Brewer","10-07-1971", "OTHER", 'ODG88UJU0DJ'),
  ("MZJ39NBP7DU","Nell Blackwell","01-06-1984", "MALE", 'HFI61GYQ6MU');
```

TRADERS TABLE



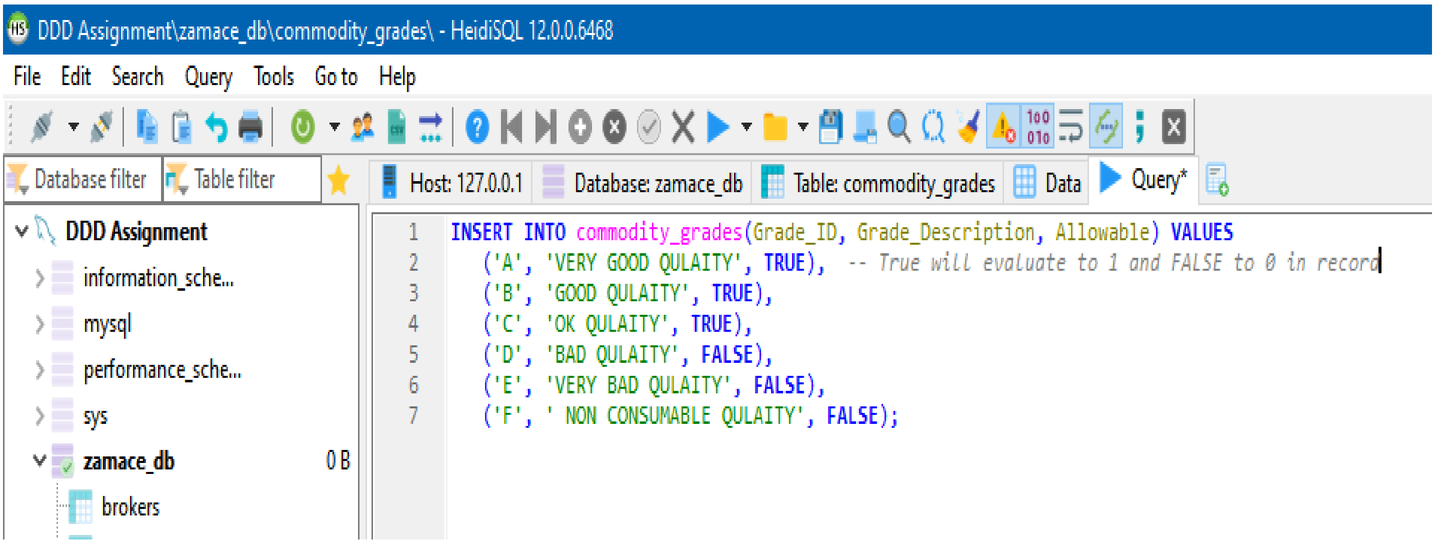
```sql
INSERT INTO traders (Trader_ID, Trader_Name, DOB, Gender, Broker_ID) VALUES
 ("OLE64WCC4FR", "Rosalyn Garcia", "1990-08-14", "FEMALE", 'FKS65LMV7GF'),
 ("WKW52WTE3UV", "Katelyn Atkinson", "1974-02-18", "FEMALE", 'GHU78GDC5HV'),
 ("KOU42IGR5FO", "Sopoline Munoz", "1991-05-26", "MALE", 'REW46SWP4CT'),
 ("DKC30OSC9KI", "Shad Torres", "1976-01-12", "MALE", 'YWU15ZUU4UU'),
 ("LPM24YLZ8CY", "Silas Clark", "1993-02-20", "MALE", 'HOQ17HOG3BU'),
 ("LRQ30FWD1HP", "Amy Potts", "1980-09-29", "MALE", 'YWU15ZUU4UU'),
 ("HOK17UMQ5RK", "Desiree Morrison", "1997-03-20", "FEMALE", 'HOQ17HOG3BU'),
 ("ZTG42OXP1KC", "Maryam Vincent", "1996-05-17", "MALE", 'JOB14WAP7LG'),
 ("LCG25PFR8CI", "Nichole Noble", "1973-06-27", "FEMALE", 'JOB14WAP7LG'),
 ("OSW02KDY1XW", "Trevor Hobbs", "1999-09-24", "FEMALE", 'GHU78GDC5HV'),
 ("BTT25NHC4LL", "Stacey Gill", "1997-03-16", "MALE", 'WYK95ROV1MB'),
 ("TIQ42EPO9PL", "Sylvia Mooney", "1992-03-12", "FEMALE", 'HRX89NUU9VR'),
 ("GOV47IFG5DY", "Breanna Compton", "1984-05-03", "OTHER", 'WYK95ROV1MB'),
 ("HRM21MDD2EF", "Sawyer Rivera", "1997-05-03", "FEMALE", 'ODG88UJU0DJ'),
 ("EGE01REK1RM", "Signe Dixon", "1996-01-08", "OTHER", 'FKS65LMV7GF'),
 ("BVV21SLN6AT", "Selma Zimmerman", "1997-07-07", "FEMALE", 'WYK95ROV1MB'),
 ("CSG46EVX4TI", "Dustin Stewart", "1985-06-10", "OTHER", 'HFI61GYQ6MU'),
 ("GYZ75BWV8XR", "Unity Holder", "1975-04-06", "MALE", 'JOB14WAP7LG'),
 ("MQY45IYG7HR", "Dennis Shields", "1971-07-10", "OTHER", 'ODG88UJU0DJ'),
 ("EXA36CXF7YG", "Mira West", "1984-06-01", "MALE", 'HFI61GYQ6MU');
```

WAREHOUSE OPERATORS TABLE



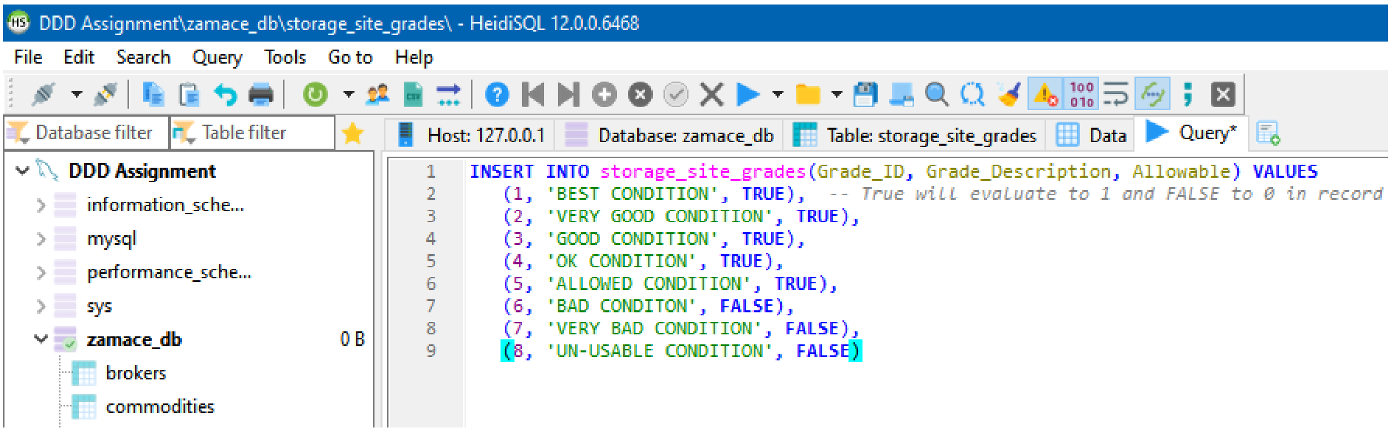
```sql
INSERT INTO wh_operators(Wh_Operator_ID, Wh_Operator_Name, Wh_Operator_Location) VALUES
  ("RBH56PGX6LA","Quam Curabitur Limited", "Plot 9169 A Nanshila Rd, Kalundu"),
  ("KQJ87EVX2BH","Elementum Purus Corporation", "Alick Nkhata Rd, P.O.Box 30175"),
  ("QUI65SKD2NO","Feugiat Company", "Plot 6300, Chinika, P.O.Box 31155"),
  ("WRG41XUT4BL","Iaculis Industries", "Buyantanshi Rd, Industrial Area, P.O.Box 31162"),
  ("VMQ53QTX9IQ","Egestas Aliquam Ltd", "2nd Shaft Chibuluma Business Park, P.O.Box"),
  ("NMH67XQW3TW","Nunc Interdum Feugiat Institute", "Kabengele Ave, Martindale, P.O.Box 20149"),
  ("RSJ89FDL4WH","Luctus LLC", "President Rd, P.O.Box 710196");
```

COMMODITY GRADES TABLE



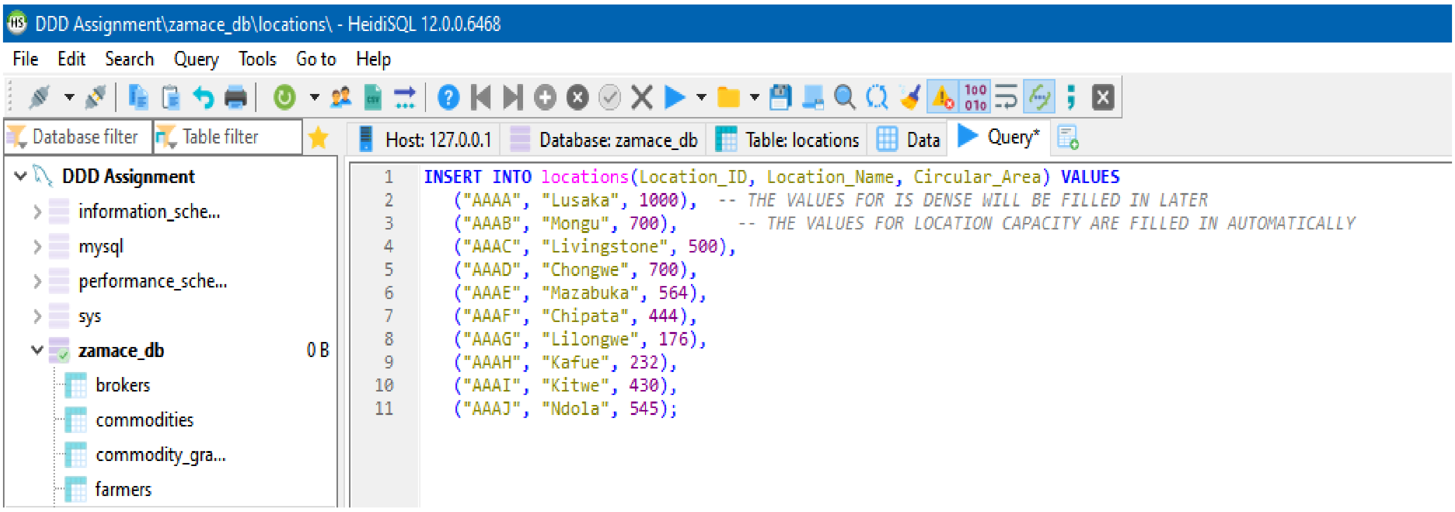
```sql
INSERT INTO commodity_grades(Grade_ID, Grade_Description, Allowable) VALUES
   ('A', 'VERY GOOD QULAITY', TRUE),  -- True will evaluate to 1 and FALSE to 0 in record
   ('B', 'GOOD QULAITY', TRUE),
   ('C', 'OK QULAITY', TRUE),
   ('D', 'BAD QULAITY', FALSE),
   ('E', 'VERY BAD QULAITY', FALSE),
   ('F', ' NON CONSUMABLE QULAITY', FALSE);
```

STORAGE SITE GRADES TABLE



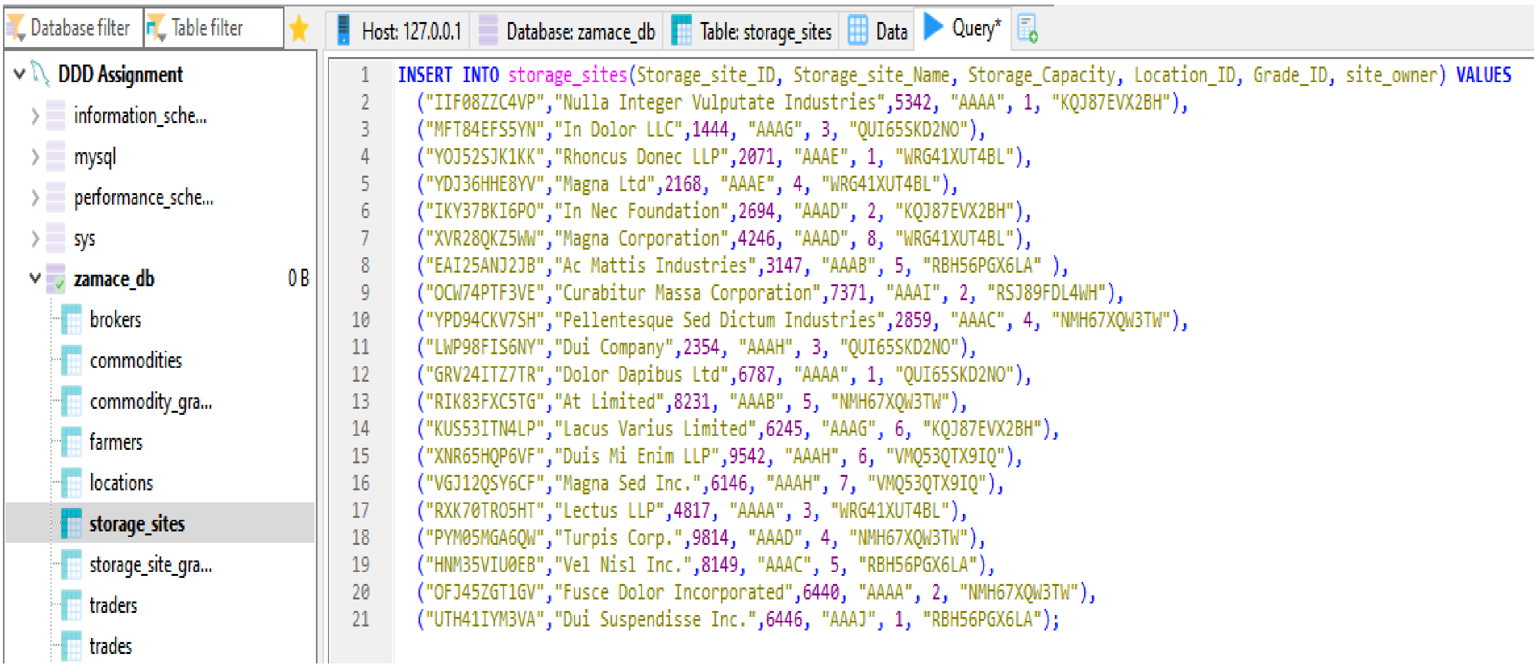
```
INSERT INTO storage_site_grades(Grade_ID, Grade_Description, Allowable) VALUES
    (1, 'BEST CONDITION', TRUE),  -- True will evaluate to 1 and FALSE to 0 in record
    (2, 'VERY GOOD CONDITION', TRUE),
    (3, 'GOOD CONDITION', TRUE),
    (4, 'OK CONDITION', TRUE),
    (5, 'ALLOWED CONDITION', TRUE),
    (6, 'BAD CONDITON', FALSE),
    (7, 'VERY BAD CONDITION', FALSE),
    (8, 'UN-USABLE CONDITION', FALSE)
```

LOCATIONS TABLE



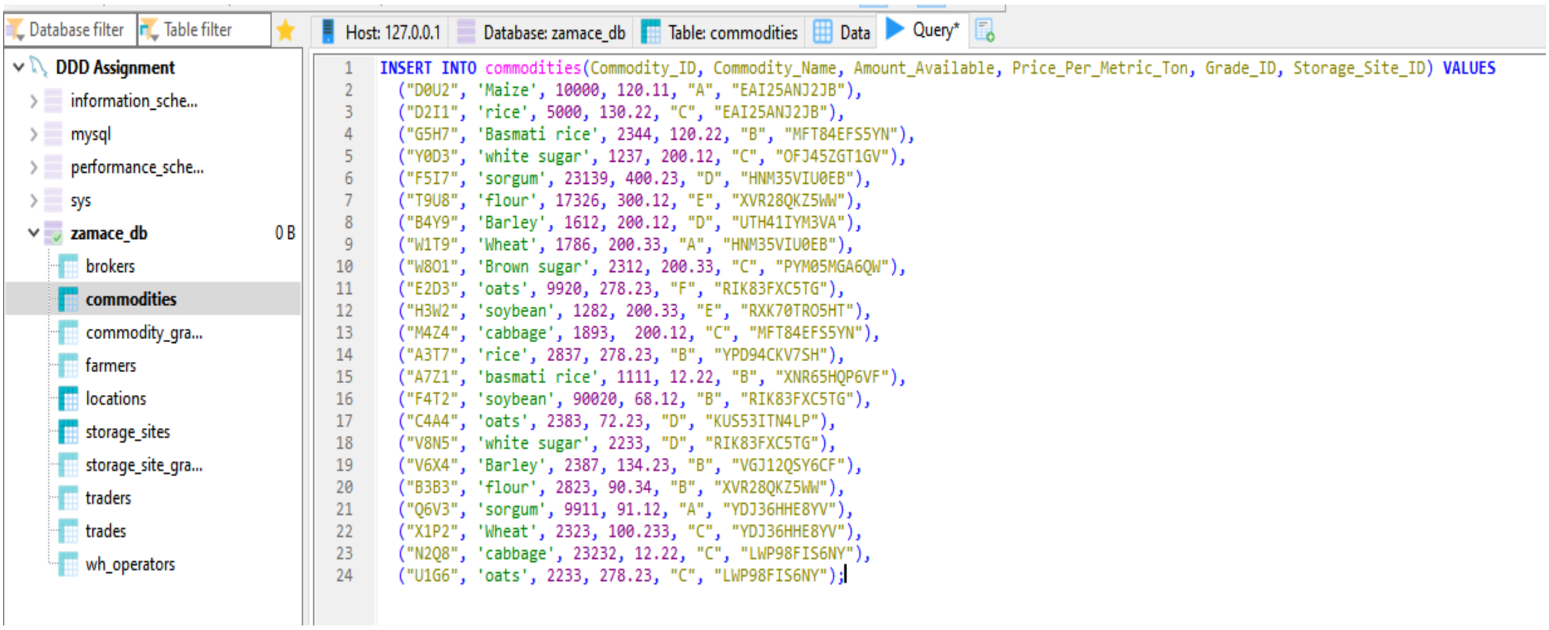
```
INSERT INTO locations(Location_ID, Location_Name, Circular_Area) VALUES
   ("AAAA", "Lusaka", 1000),  -- THE VALUES FOR IS DENSE WILL BE FILLED IN LATER
   ("AAAB", "Mongu", 700),      -- THE VALUES FOR LOCATION CAPACITY ARE FILLED IN AUTOMATICALLY
   ("AAAC", "Livingstone", 500),
   ("AAAD", "Chongwe", 700),
   ("AAAE", "Mazabuka", 564),
   ("AAAF", "Chipata", 444),
   ("AAAG", "Lilongwe", 176),
   ("AAAH", "Kafue", 232),
   ("AAAI", "Kitwe", 430),
   ("AAAJ", "Ndola", 545);
```

STORAGE SITES TABLE



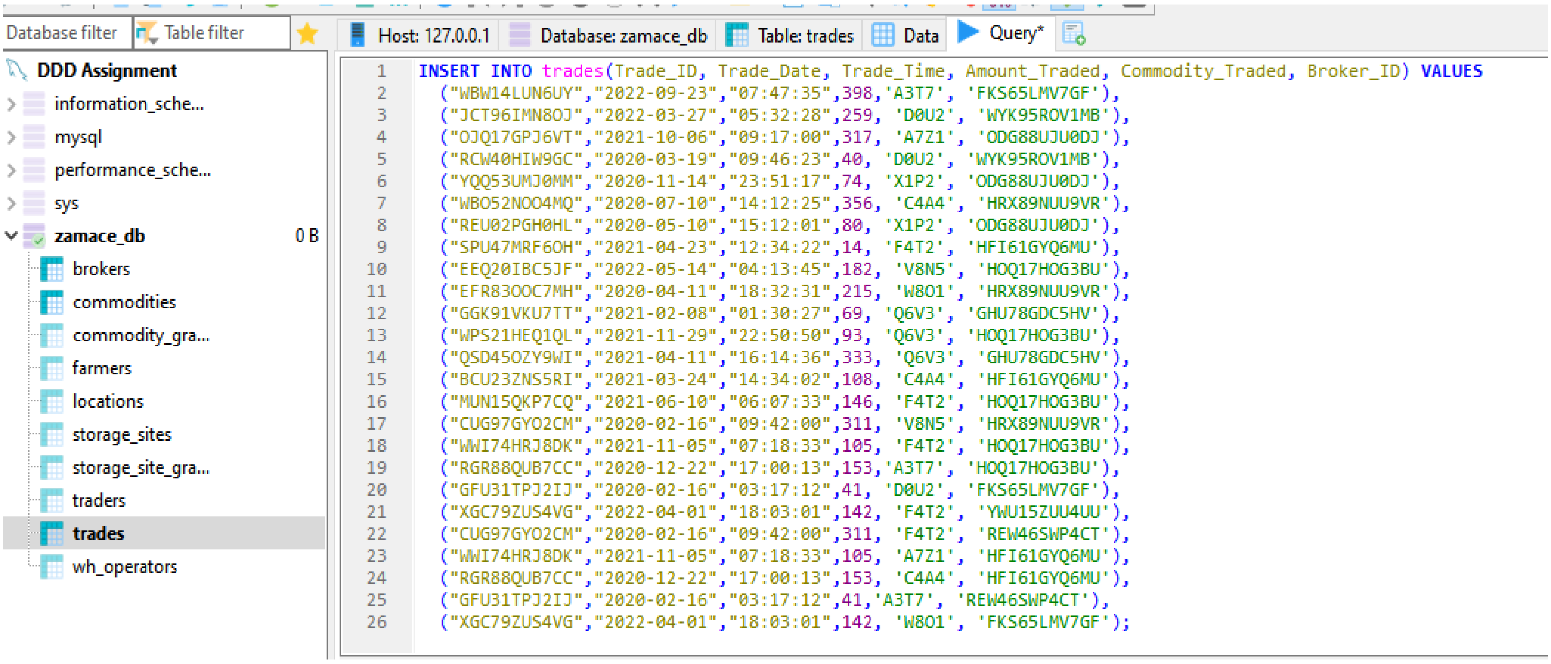
```
INSERT INTO storage_sites(Storage_site_ID, Storage_site_Name, Storage_Capacity, Location_ID, Grade_ID, site_owner) VALUES
  ("IIF08ZZC4VP","Nulla Integer Vulputate Industries",5342, "AAAA", 1, "KQJ87EVX2BH"),
  ("MFT84EFS5YN","In Dolor LLC",1444, "AAAG", 3, "QUI65SKD2NO"),
  ("YOJ52SJK1KK","Rhoncus Donec LLP",2071, "AAAE", 1, "WRG41XUT4BL"),
  ("YDJ36HHE8YV","Magna Ltd",2168, "AAAE", 4, "WRG41XUT4BL"),
  ("IKY37BKI6PO","In Nec Foundation",2694, "AAAD", 2, "KQJ87EVX2BH"),
  ("XVR28QKZ5WW","Magna Corporation",4246, "AAAD", 8, "WRG41XUT4BL"),
  ("EAI25ANJ2JB","Ac Mattis Industries",3147, "AAAB", 5, "RBH56PGX6LA" ),
  ("OCW74PTF3VE","Curabitur Massa Corporation",7371, "AAAI", 2, "RSJ89FDL4WH"),
  ("YPD94CKV7SH","Pellentesque Sed Dictum Industries",2859, "AAAC", 4, "NMH67XQW3TW"),
  ("LWP98FIS6NY","Dui Company",2354, "AAAH", 3, "QUI65SKD2NO"),
  ("GRV24ITZ7TR","Dolor Dapibus Ltd",6787, "AAAA", 1, "QUI65SKD2NO"),
  ("RIK83FXC5TG","At Limited",8231, "AAAB", 5, "NMH67XQW3TW"),
  ("KUS53ITN4LP","Lacus Varius Limited",6245, "AAAG", 6, "KQJ87EVX2BH"),
  ("XNR65HQP6VF","Duis Mi Enim LLP",9542, "AAAH", 6, "VMQ53QTX9IQ"),
  ("VGJ12QSY6CF","Magna Sed Inc.",6146, "AAAH", 7, "VMQ53QTX9IQ"),
  ("RXK70TRO5HT","Lectus LLP",4817, "AAAA", 3, "WRG41XUT4BL"),
  ("PYM05MGA6QW","Turpis Corp.",9814, "AAAD", 4, "NMH67XQW3TW"),
  ("HNM35VIU0EB","Vel Nisl Inc.",8149, "AAAC", 5, "RBH56PGX6LA"),
  ("OFJ45ZGT1GV","Fusce Dolor Incorporated",6440, "AAAA", 2, "NMH67XQW3TW"),
  ("UTH41IYM3VA","Dui Suspendisse Inc.",6446, "AAAJ", 1, "RBH56PGX6LA");
```

COMMODITIES TABLE



```
INSERT INTO commodities(Commodity_ID, Commodity_Name, Amount_Available, Price_Per_Metric_Ton, Grade_ID, Storage_Site_ID) VALUES
  ("D0U2", 'Maize', 10000, 120.11, "A", "EAI25ANJ2JB"),
  ("D2I1", 'rice', 5000, 130.22, "C", "EAI25ANJ2JB"),
  ("G5H7", 'Basmati rice', 2344, 120.22, "B", "MFT84EFS5YN"),
  ("Y0D3", 'white sugar', 1237, 200.12, "C", "OFJ45ZGT1GV"),
  ("F5I7", 'sorgum', 23139, 400.23, "D", "HNM35VIU0EB"),
  ("T9U8", 'flour', 17326, 300.12, "E", "XVR28QKZ5WW"),
  ("B4Y9", 'Barley', 1612, 200.12, "D", "UTH41IYM3VA"),
  ("W1T9", 'Wheat', 1786, 200.33, "A", "HNM35VIU0EB"),
  ("W8O1", 'Brown sugar', 2312, 200.33, "C", "PYM05MGA6QW"),
  ("E2D3", 'oats', 9920, 278.23, "F", "RIK83FXC5TG"),
  ("H3W2", 'soybean', 1282, 200.33, "E", "RXK70TRO5HT"),
  ("M4Z4", 'cabbage', 1893,  200.12, "C", "MFT84EFS5YN"),
  ("A3T7", 'rice', 2837, 278.23, "B", "YPD94CKV7SH"),
  ("A7Z1", 'basmati rice', 1111, 12.22, "B", "XNR65HQP6VF"),
  ("F4T2", 'soybean', 90020, 68.12, "B", "RIK83FXC5TG"),
  ("C4A4", 'oats', 2383, 72.23, "D", "KUS53ITN4LP"),
  ("V8N5", 'white sugar', 2233, "D", "RIK83FXC5TG"),
  ("V6X4", 'Barley', 2387, 134.23, "B", "VGJ12QSY6CF"),
  ("B3B3", 'flour', 2823, 90.34, "B", "XVR28QKZ5WW"),
  ("Q6V3", 'sorgum', 9911, 91.12, "A", "YDJ36HHE8YV"),
  ("X1P2", 'Wheat', 2323, 100.233, "C", "YDJ36HHE8YV"),
  ("N2Q8", 'cabbage', 23232, 12.22, "C", "LWP98FIS6NY"),
  ("U1G6", 'oats', 2233, 278.23, "C", "LWP98FIS6NY");
```

TRADES TABLE



```
INSERT INTO trades(Trade_ID, Trade_Date, Trade_Time, Amount_Traded, Commodity_Traded, Broker_ID) VALUES
  ("WBW14LUN6UY","2022-09-23","07:47:35",398,'A3T7', 'FKS65LMV7GF'),
  ("JCT96IMN8OJ","2022-03-27","05:32:28",259, 'D0U2', 'WYK95ROV1MB'),
  ("OJQ17GPJ6VT","2021-10-06","09:17:00",317, 'A7Z1', 'ODG88UJU0DJ'),
  ("RCW40HIW9GC","2020-03-19","09:46:23",40, 'D0U2', 'WYK95ROV1MB'),
  ("YQQ53UMJ0MM","2020-11-14","23:51:17",74, 'X1P2', 'ODG88UJU0DJ'),
  ("WBO52NOO4MQ","2020-07-10","14:12:25",356, 'C4A4', 'HRX89NUU9VR'),
  ("REU02PGH0HL","2020-05-10","15:12:01",80, 'X1P2', 'ODG88UJU0DJ'),
  ("SPU47MRF6OH","2021-04-23","12:34:22",14, 'F4T2', 'HFI61GYQ6MU'),
  ("EEQ20IBC5JF","2022-05-14","04:13:45",182, 'V8N5', 'HOQ17HOG3BU'),
  ("EFR83OOC7MH","2020-04-11","18:32:31",215, 'W8O1', 'HRX89NUU9VR'),
  ("GGK91VKU7TT","2021-02-08","01:30:27",69, 'Q6V3', 'GHU78GDC5HV'),
  ("WPS21HEQ1QL","2021-11-29","22:50:50",93, 'Q6V3', 'HOQ17HOG3BU'),
  ("QSD45OZY9WI","2021-04-11","16:14:36",333, 'Q6V3', 'GHU78GDC5HV'),
  ("BCU23ZNS5RI","2021-03-24","14:34:02",108, 'C4A4', 'HFI61GYQ6MU'),
  ("MUN15QKP7CQ","2021-06-10","06:07:33",146, 'F4T2', 'HOQ17HOG3BU'),
  ("CUG97GYO2CM","2020-02-16","09:42:00",311, 'V8N5', 'HRX89NUU9VR'),
  ("WWI74HRJ8DK","2021-11-05","07:18:33",105, 'F4T2', 'HOQ17HOG3BU'),
  ("RGR88QUB7CC","2020-12-22","17:00:13",153,'A3T7', 'HOQ17HOG3BU'),
  ("GFU31TPJ2IJ","2020-02-16","03:17:12",41, 'D0U2', 'FKS65LMV7GF'),
  ("XGC79ZUS4VG","2022-04-01","18:03:01",142, 'F4T2', 'YWU15ZUU4UU'),
  ("CUG97GYO2CM","2020-02-16","09:42:00",311, 'F4T2', 'REW46SWP4CT'),
  ("WWI74HRJ8DK","2021-11-05","07:18:33",105, 'A7Z1', 'HFI61GYQ6MU'),
  ("RGR88QUB7CC","2020-12-22","17:00:13",153, 'C4A4', 'HFI61GYQ6MU'),
  ("GFU31TPJ2IJ","2020-02-16","03:17:12",41,'A3T7', 'REW46SWP4CT'),
  ("XGC79ZUS4VG","2022-04-01","18:03:01",142, 'W8O1', 'FKS65LMV7GF');
```

How I developed the scripts, order or running and errors I encountered:

I developed the scripts using relevant data, I used online data generators to help me come up with large sums of relevant data in order to ease the process of inserting data. Then I simply took the data and placed it into the insert statement.
The order of running the scripts was the same as the order of creating the tables. I entered data in tables with no foreign keys first and then went to the other tables.
The issues I encountered with that I had to copy foreign keys in the foreign key columns and I had to make sure I copied each foreign key correctly otherwise the record would not get entered.

TASK (7) (Enhancement using SQL)

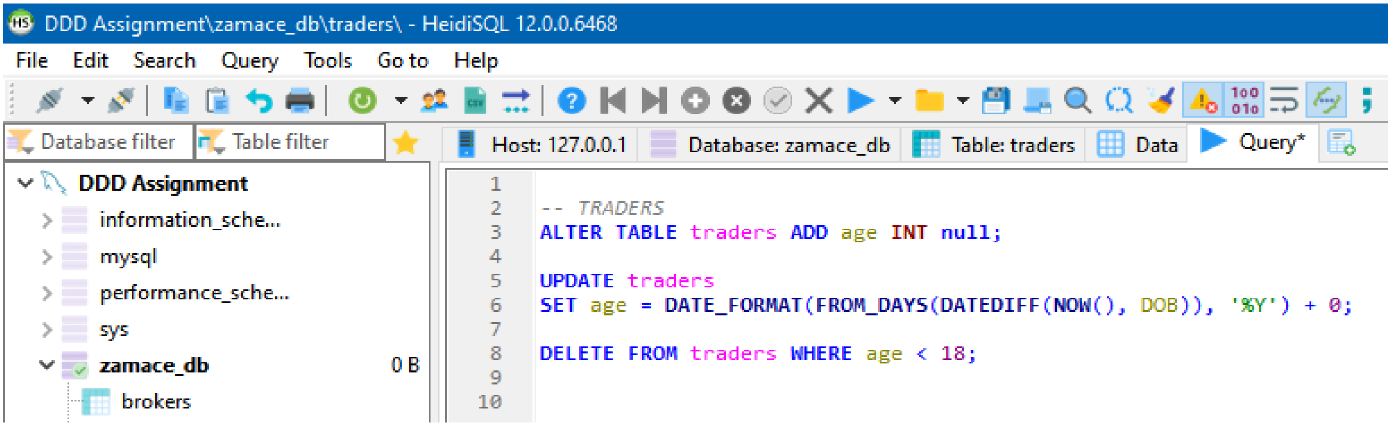
There a few potential changes that could be made to the original database design. Namely;

- We can add an AGE column to the farmers and trader's table
- We can add a check constraint to the trades table on the amount traded column so that the amount traded is not greater than the amount available in the commodities table.
- Add a column in location table which gets number of storage sites in that location.

The reason for the first change is that it will become easier for companies to check the age of their users and only allow users above age 18.

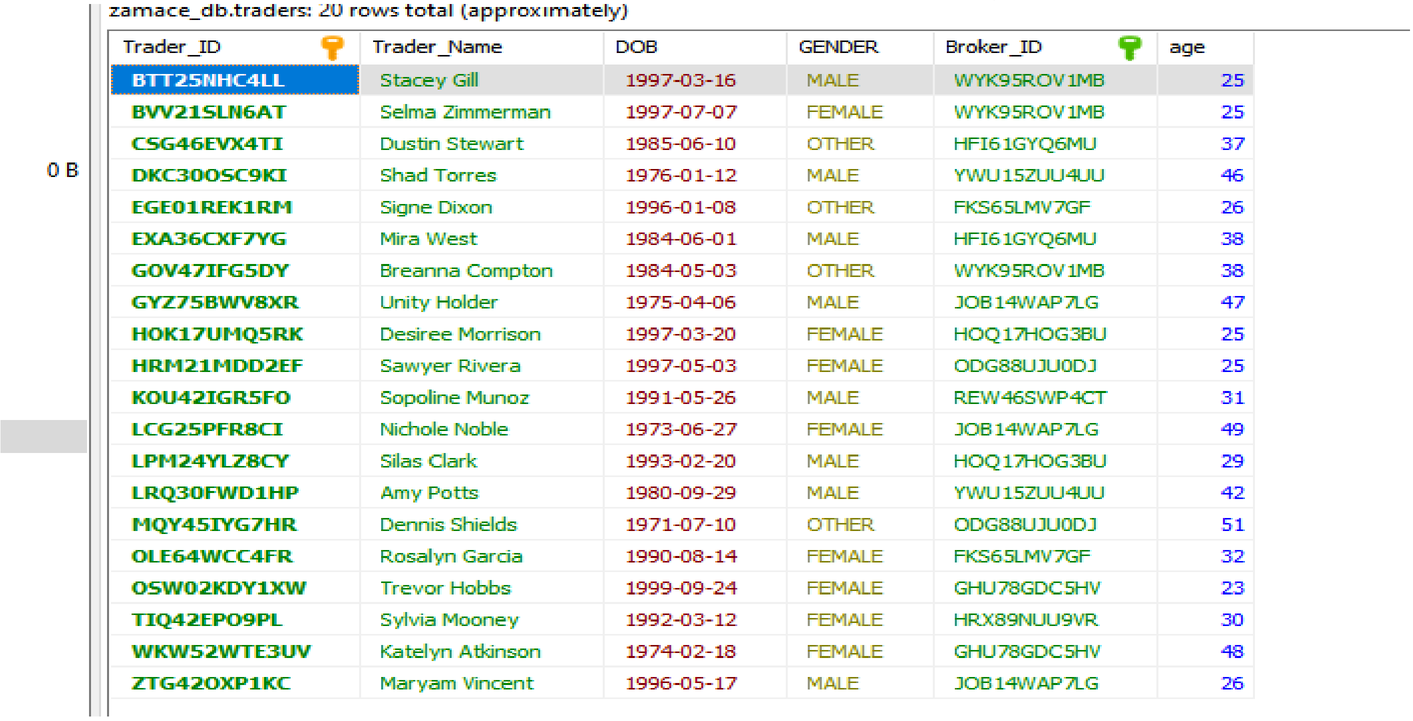
The reason for the second change is brokers will only be able to trade less than the amount of a certain commodity that is available in the storage site otherwise it will give an error.

Using SQL to implement first change: (Same done for farmers table).



```

-- TRADERS
ALTER TABLE traders ADD age INT null;

UPDATE traders
SET age = DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(), DOB)), '%Y') + 0;

DELETE FROM traders WHERE age < 18;

```

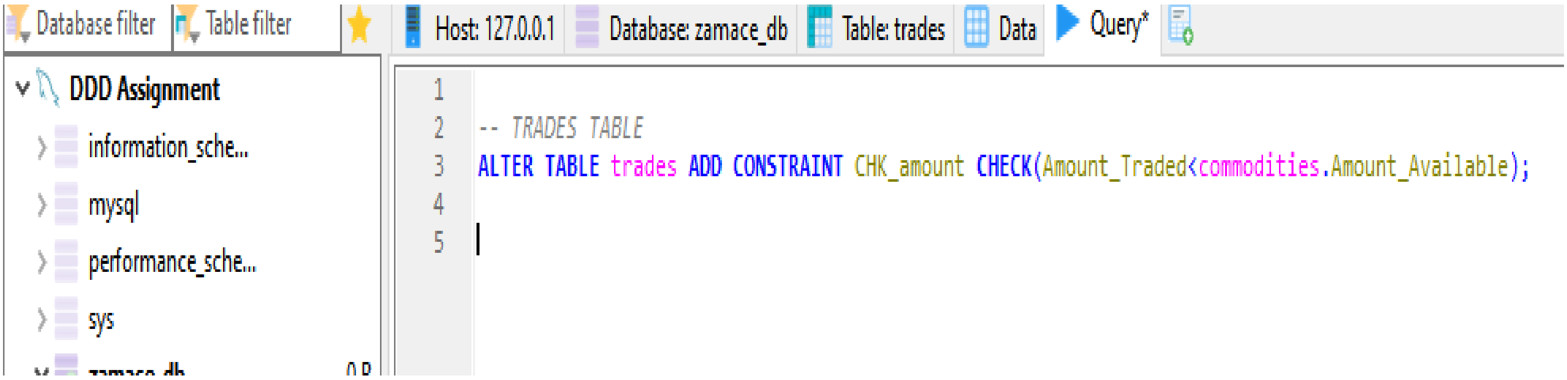
RESULTS OF FIRST CHANGE (Similar results for farmers table)



zamace_db.traders: 20 rows total (approximately)

Trader_ID	Trader_Name	DOB	GENDER	Broker_ID	age
BTT25NHC4LL	Stacey Gill	1997-03-16	MALE	WYK95ROV1MB	25
BVV21SLN6AT	Selma Zimmerman	1997-07-07	FEMALE	WYK95ROV1MB	25
CSG46EVX4TI	Dustin Stewart	1985-06-10	OTHER	HFI61GYQ6MU	37
DKC30OSC9KI	Shad Torres	1976-01-12	MALE	YWU15ZUU4UU	46
EGE01REK1RM	Signe Dixon	1996-01-08	OTHER	FKS65LMV7GF	26
EXA36CXF7YG	Mira West	1984-06-01	MALE	HFI61GYQ6MU	38
GOV47IFG5DY	Breanna Compton	1984-05-03	OTHER	WYK95ROV1MB	38
GYZ75BWV8XR	Unity Holder	1975-04-06	MALE	JOB14WAP7LG	47
HOK17UMQ5RK	Desiree Morrison	1997-03-20	FEMALE	HOQ17HOG3BU	25
HRM21MDD2EF	Sawyer Rivera	1997-05-03	FEMALE	ODG88UJU0DJ	25
KOU42IGR5FO	Sopoline Munoz	1991-05-26	MALE	REW46SWP4CT	31
LCG25PFR8CI	Nichole Noble	1973-06-27	FEMALE	JOB14WAP7LG	49
LPM24YLZ8CY	Silas Clark	1993-02-20	MALE	HOQ17HOG3BU	29
LRQ30FWD1HP	Amy Potts	1980-09-29	MALE	YWU15ZUU4UU	42
MQY45IYG7HR	Dennis Shields	1971-07-10	OTHER	ODG88UJU0DJ	51
OLE64WCC4FR	Rosalyn Garcia	1990-08-14	FEMALE	FKS65LMV7GF	32
OSW02KDY1XW	Trevor Hobbs	1999-09-24	FEMALE	GHU78GDC5HV	23
TIQ42EPO9PL	Sylvia Mooney	1992-03-12	FEMALE	HRX89NUU9VR	30
WKW52WTE3UV	Katelyn Atkinson	1974-02-18	FEMALE	GHU78GDC5HV	48
ZTG42OXP1KC	Maryam Vincent	1996-05-17	MALE	JOB14WAP7LG	26

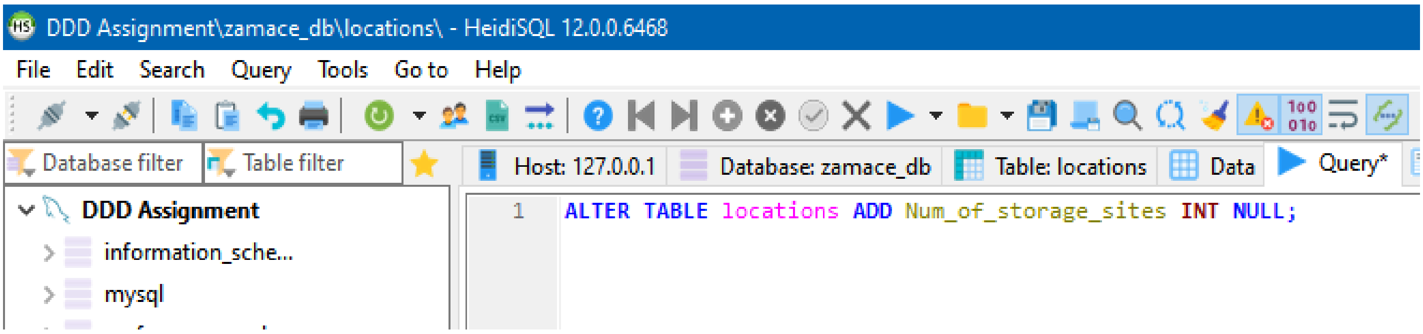
Using SQL To implement second change;



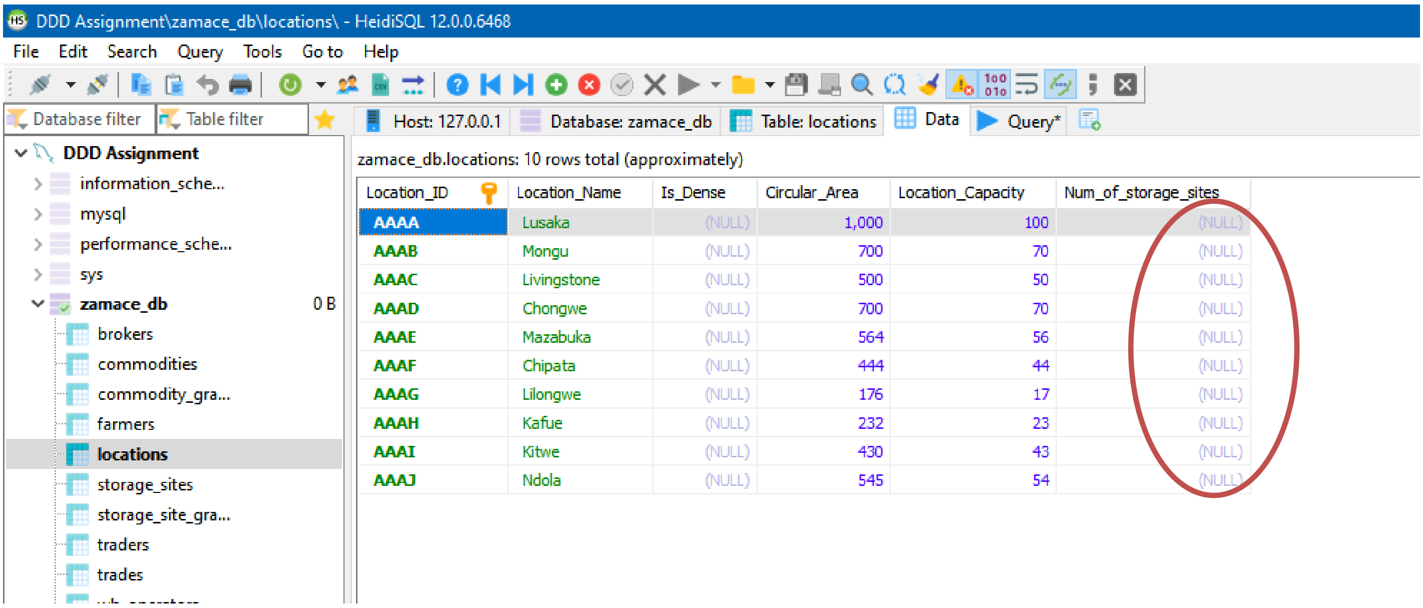
```
-- TRADES TABLE
ALTER TABLE trades ADD CONSTRAINT CHK_amount CHECK(Amount_Traded<commodities.Amount_Available);
```

By adding this constraint to the trades table, we can now only allow trades that have the amount traded less than the amount of that commodity available,

Using SQL to implement third change.



```
ALTER TABLE locations ADD Num_of_storage_sites INT NULL;
```



zamace_db.locations: 10 rows total (approximately)

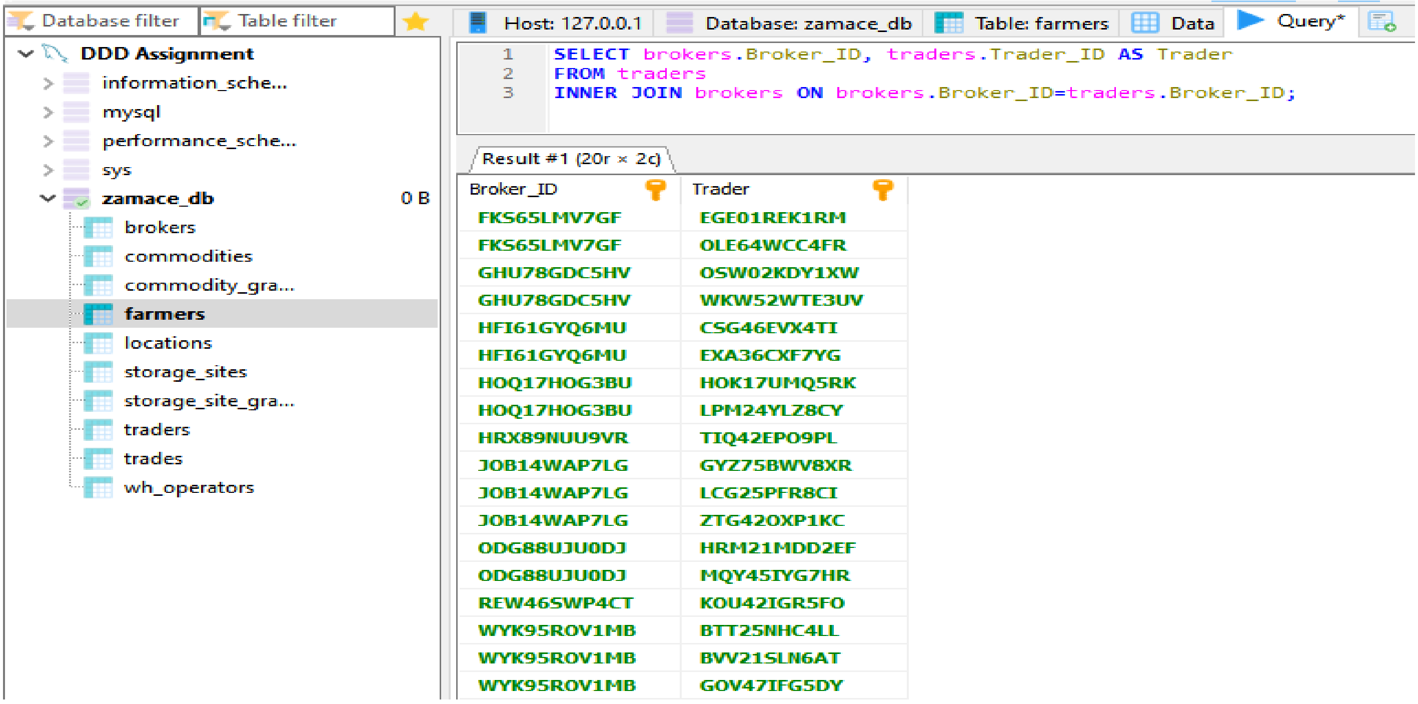
Location_ID	Location_Name	Is_Dense	Circular_Area	Location_Capacity	Num_of_storage_sites
AAAA	Lusaka	(NULL)	1,000	100	(NULL)
AAAB	Mongu	(NULL)	700	70	(NULL)
AAAC	Livingstone	(NULL)	500	50	(NULL)
AAAD	Chongwe	(NULL)	700	70	(NULL)
AAAE	Mazabuka	(NULL)	564	56	(NULL)
AAAF	Chipata	(NULL)	444	44	(NULL)
AAAG	Lilongwe	(NULL)	176	17	(NULL)
AAAH	Kafue	(NULL)	232	23	(NULL)
AAAI	Kitwe	(NULL)	430	43	(NULL)
AAAJ	Ndola	(NULL)	545	54	(NULL)

TASK (8) (SQL REPORTS)

Query 1:

This query will inner join the brokers and traders table on condition that the traders broker id is = brokers broker id We want to know the brokers each trader use to place trades on ZAMACE, this way we can catch people who are using brokers who are not registered with ZAMACE. This same query can be done on farmers table as well.

SQL QUERY WITH ITS RUNTIME RESULTS ON THE BOTTOM IN GREEN:



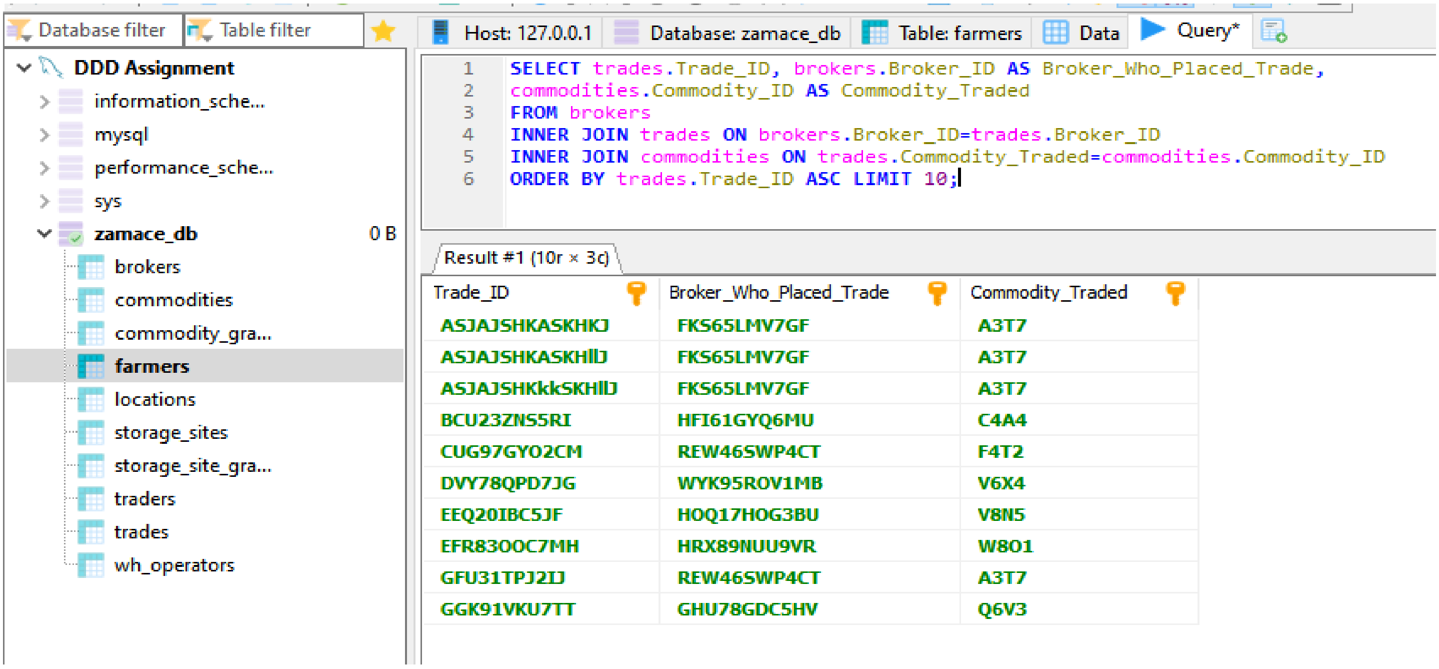
```sql
SELECT brokers.Broker_ID, traders.Trader_ID AS Trader
FROM traders
INNER JOIN brokers ON brokers.Broker_ID=traders.Broker_ID;
```

Result #1 (20r × 2c)

Broker_ID	Trader
FKS65LMV7GF	EGE01REK1RM
FKS65LMV7GF	OLE64WCC4FR
GHU78GDC5HV	OSW02KDY1XW
GHU78GDC5HV	WKW52WTE3UV
HFI61GYQ6MU	CSG46EVX4TI
HFI61GYQ6MU	EXA36CXF7YG
HOQ17HOG3BU	HOK17UMQ5RK
HOQ17HOG3BU	LPM24YLZ8CY
HRX89NUU9VR	TIQ42EPO9PL
JOB14WAP7LG	GYZ75BWV8XR
JOB14WAP7LG	LCG25PFR8CI
JOB14WAP7LG	ZTG42OXP1KC
ODG88UJU0DJ	HRM21MDD2EF
ODG88UJU0DJ	MQY45IYG7HR
REW46SWP4CT	KOU42IGR5FO
WYK95ROV1MB	BTT25NHC4LL
WYK95ROV1MB	BVV21SLN6AT
WYK95ROV1MB	GOV47IFG5DY

Query2:

In this query we will retrieve the first 10 trades that were placed by brokers, and what commodity was traded in each trade. This can be useful if we want to know what was traded by a certain broker.



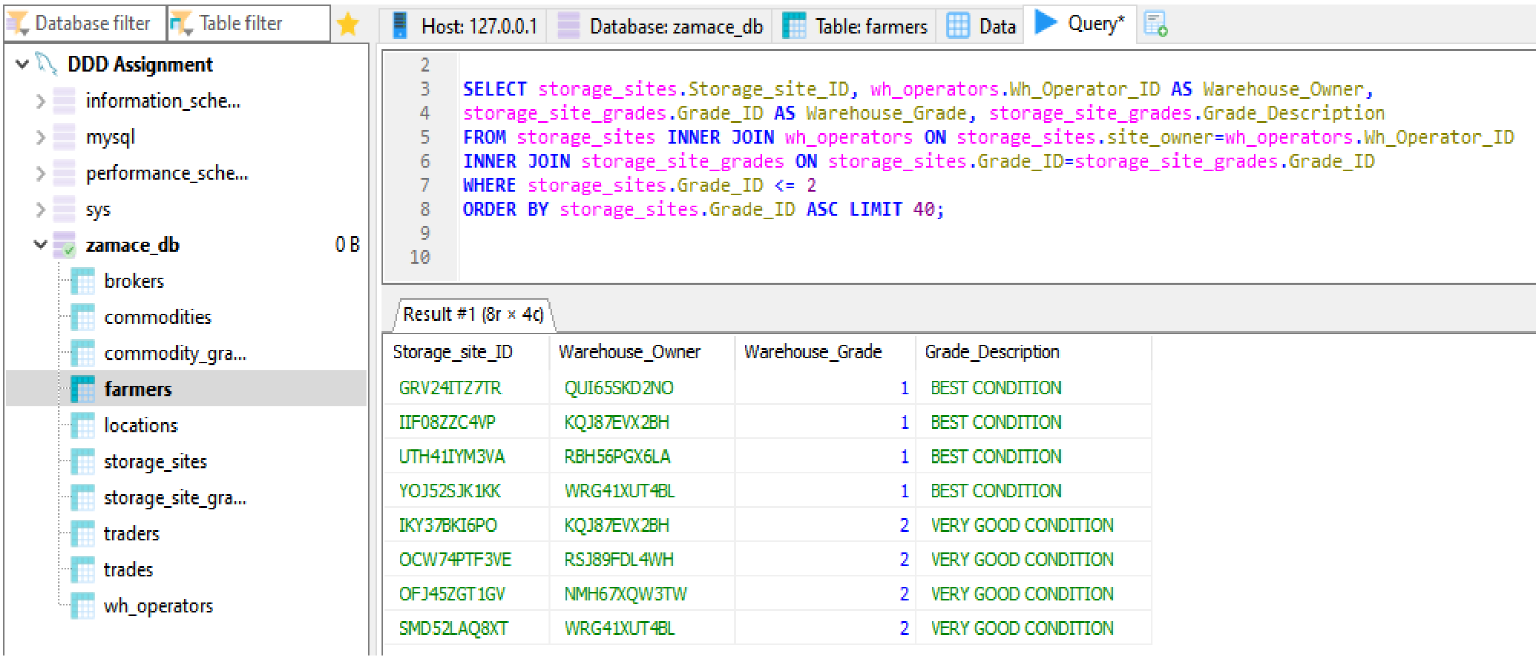
```sql
SELECT trades.Trade_ID, brokers.Broker_ID AS Broker_Who_Placed_Trade,
commodities.Commodity_ID AS Commodity_Traded
FROM brokers
INNER JOIN trades ON brokers.Broker_ID=trades.Broker_ID
INNER JOIN commodities ON trades.Commodity_Traded=commodities.Commodity_ID
ORDER BY trades.Trade_ID ASC LIMIT 10;
```

Result #1 (10r × 3c)

Trade_ID	Broker_Who_Placed_Trade	Commodity_Traded
ASJAJSHKASKHKJ	FKS65LMV7GF	A3T7
ASJAJSHKASKHlIJ	FKS65LMV7GF	A3T7
ASJAJSHKkkSKHlIJ	FKS65LMV7GF	A3T7
BCU23ZNS5RI	HFI61GYQ6MU	C4A4
CUG97GYO2CM	REW46SWP4CT	F4T2
DVY78QPD7JG	WYK95ROV1MB	V6X4
EEQ20IBC5JF	HOQ17HOG3BU	V8N5
EFR83OOC7MH	HRX89NUU9VR	W8O1
GFU31TPJ2IJ	REW46SWP4CT	A3T7
GGK91VKU7TT	GHU78GDC5HV	Q6V3

Query 3:

In this query we will retrieve the storage sites along with their owners and grades, but we only want those with grades that are 1 or 2. This query can be useful to find out the best storage sites that the ZAMACE ecosystem has to offer.



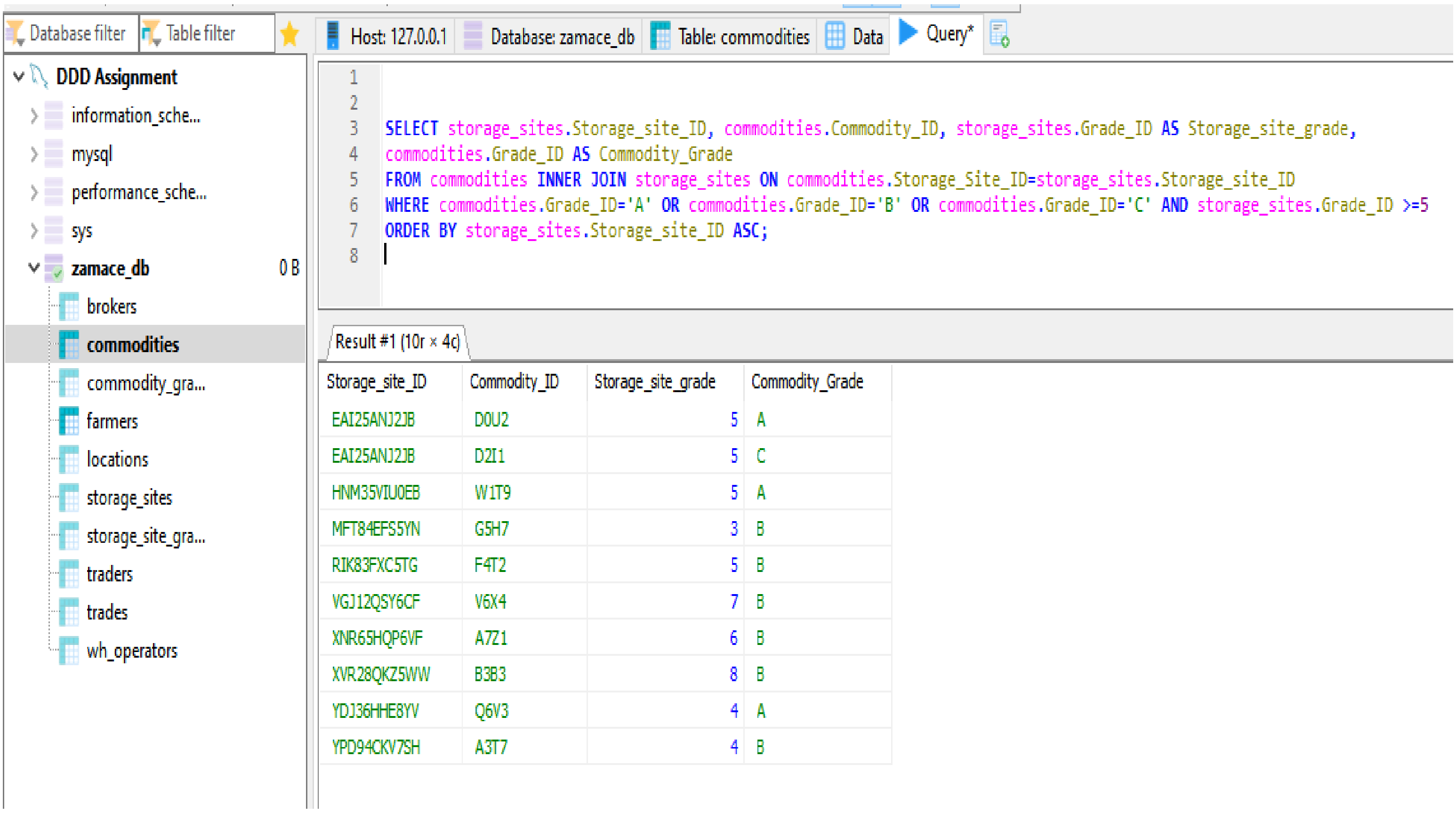
```sql
SELECT storage_sites.Storage_site_ID, wh_operators.Wh_Operator_ID AS Warehouse_Owner,
storage_site_grades.Grade_ID AS Warehouse_Grade, storage_site_grades.Grade_Description
FROM storage_sites INNER JOIN wh_operators ON storage_sites.site_owner=wh_operators.Wh_Operator_ID
INNER JOIN storage_site_grades ON storage_sites.Grade_ID=storage_site_grades.Grade_ID
WHERE storage_sites.Grade_ID <= 2
ORDER BY storage_sites.Grade_ID ASC LIMIT 40;
```

Result #1 (8r × 4c)

Storage_site_ID	Warehouse_Owner	Warehouse_Grade	Grade_Description
GRV24ITZ7TR	QUI65SKD2NO	1	BEST CONDITION
IIF08ZZC4VP	KQJ87EVX2BH	1	BEST CONDITION
UTH41IYM3VA	RBH56PGX6LA	1	BEST CONDITION
YOJ52SJK1KK	WRG41XUT4BL	1	BEST CONDITION
IKY37BKI6PO	KQJ87EVX2BH	2	VERY GOOD CONDITION
OCW74PTF3VE	RSJ89FDL4WH	2	VERY GOOD CONDITION
OFJ45ZGT1GV	NMH67XQW3TW	2	VERY GOOD CONDITION
SMD52LAQ8XT	WRG41XUT4BL	2	VERY GOOD CONDITION

Query 4:

In this query we want to retrieve the commodities present in a storage site that have a grade of A, B or C, and the storage site must be above **grade 5 (which is a bad grade).** **Only Grade 1 to 4 are good grades**. This can be helpful is the company wants to move all the good grade commodities present in the bad warehouses to the good warehouses.



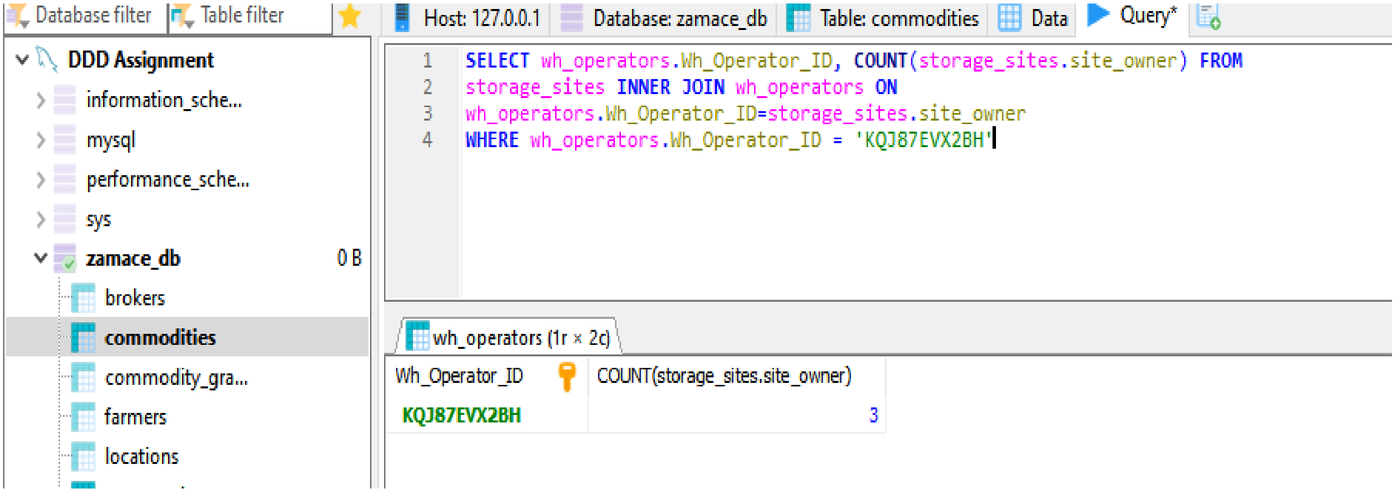
```sql
SELECT storage_sites.Storage_site_ID, commodities.Commodity_ID, storage_sites.Grade_ID AS Storage_site_grade,
commodities.Grade_ID AS Commodity_Grade
FROM commodities INNER JOIN storage_sites ON commodities.Storage_Site_ID=storage_sites.Storage_site_ID
WHERE commodities.Grade_ID='A' OR commodities.Grade_ID='B' OR commodities.Grade_ID='C' AND storage_sites.Grade_ID >=5
ORDER BY storage_sites.Storage_site_ID ASC;
```

Result #1 (10r × 4c)

Storage_site_ID	Commodity_ID	Storage_site_grade	Commodity_Grade
EAI25ANJ2JB	D0U2	5	A
EAI25ANJ2JB	D2I1	5	C
HNM35VIU0EB	W1T9	5	A
MFT84EFS5YN	G5H7	3	B
RIK83FXC5TG	F4T2	5	B
VGJ12QSY6CF	V6X4	7	B
XNR65HQP6VF	A7Z1	6	B
XVR28QKZ5WW	B3B3	8	B
YDJ36HHE8YV	Q6V3	4	A
YPD94CKV7SH	A3T7	4	B

Query 5:

We want Count how many storage sites a certain warehouse operator owns.

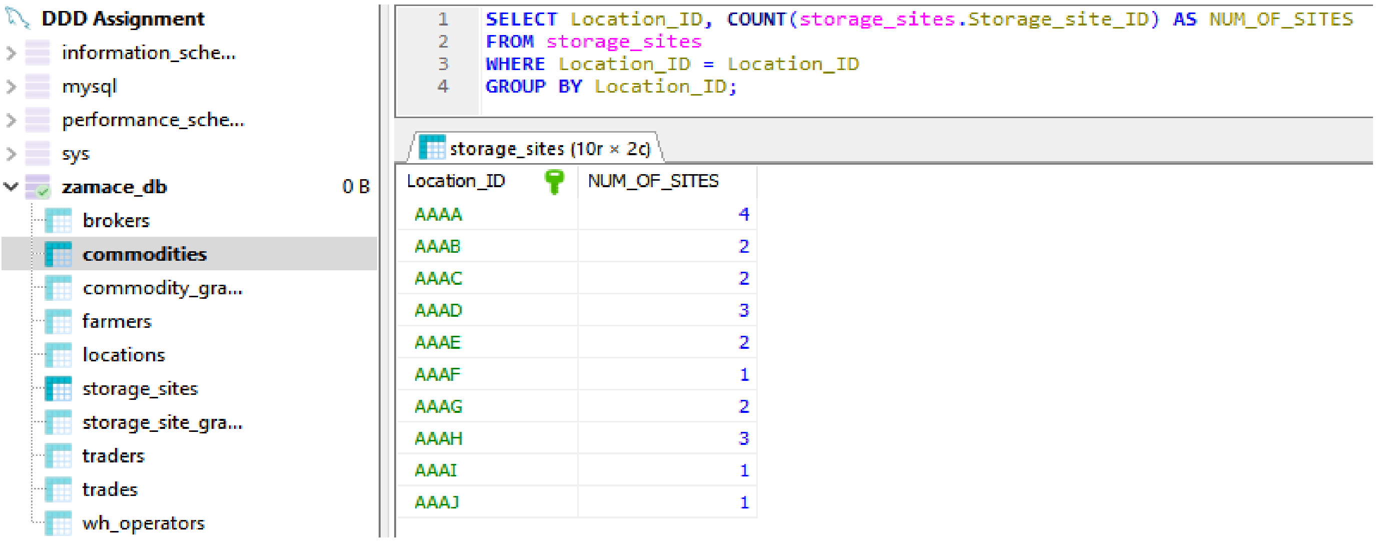


```sql
SELECT wh_operators.Wh_Operator_ID, COUNT(storage_sites.site_owner) FROM
storage_sites INNER JOIN wh_operators ON
wh_operators.Wh_Operator_ID=storage_sites.site_owner
WHERE wh_operators.Wh_Operator_ID = 'KQJ87EVX2BH'
```

Wh_Operator_ID	COUNT(storage_sites.site_owner)
KQJ87EVX2BH	3

Query 6:

We want to get the number of sites in each location present in our database;

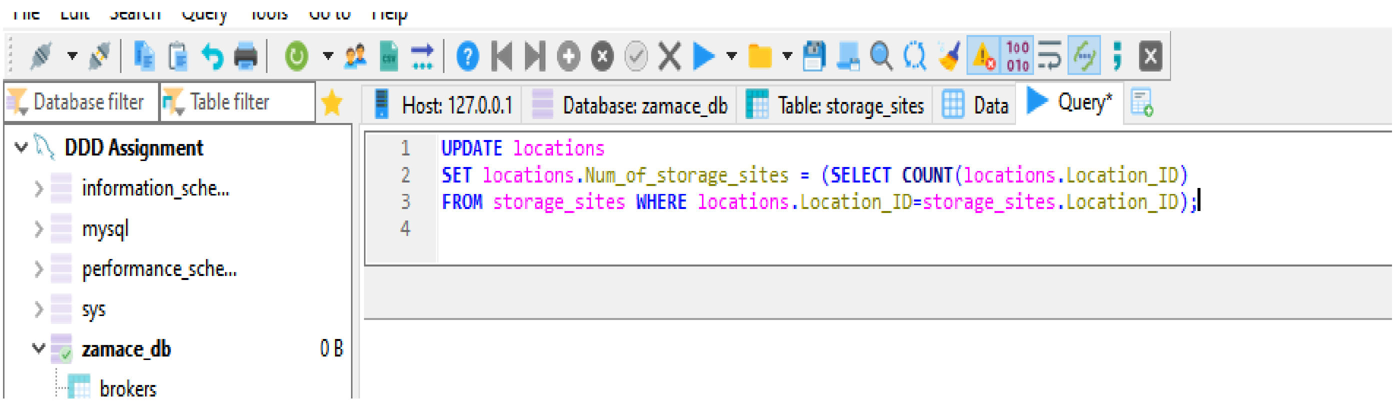


```sql
SELECT Location_ID, COUNT(storage_sites.Storage_site_ID) AS NUM_OF_SITES
FROM storage_sites
WHERE Location_ID = Location_ID
GROUP BY Location_ID;
```

Location_ID	NUM_OF_SITES
AAAA	4
AAAB	2
AAAC	2
AAAD	3
AAAE	2
AAAF	1
AAAG	2
AAAH	3
AAAI	1
AAAJ	1

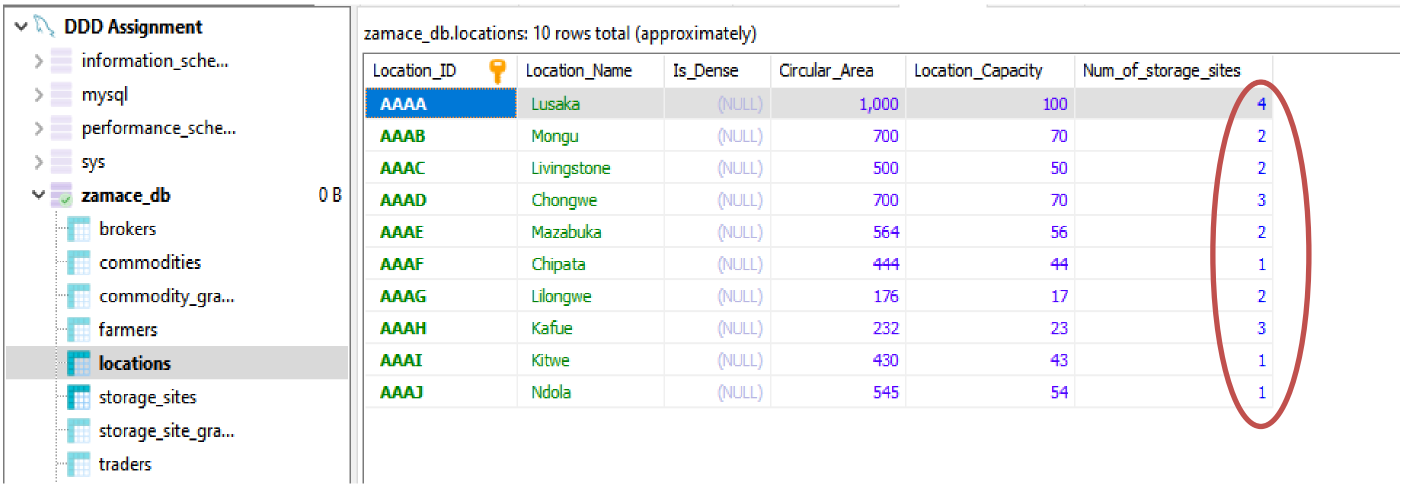
Query 7:

Since the “Num of storage sites column in the locations table had null values) we want to set those values to the actual number of storage sites in that location. This has a sub query



```sql
UPDATE locations
SET locations.Num_of_storage_sites = (SELECT COUNT(locations.Location_ID)
FROM storage_sites WHERE locations.Location_ID=storage_sites.Location_ID);
```

Query 7 results:

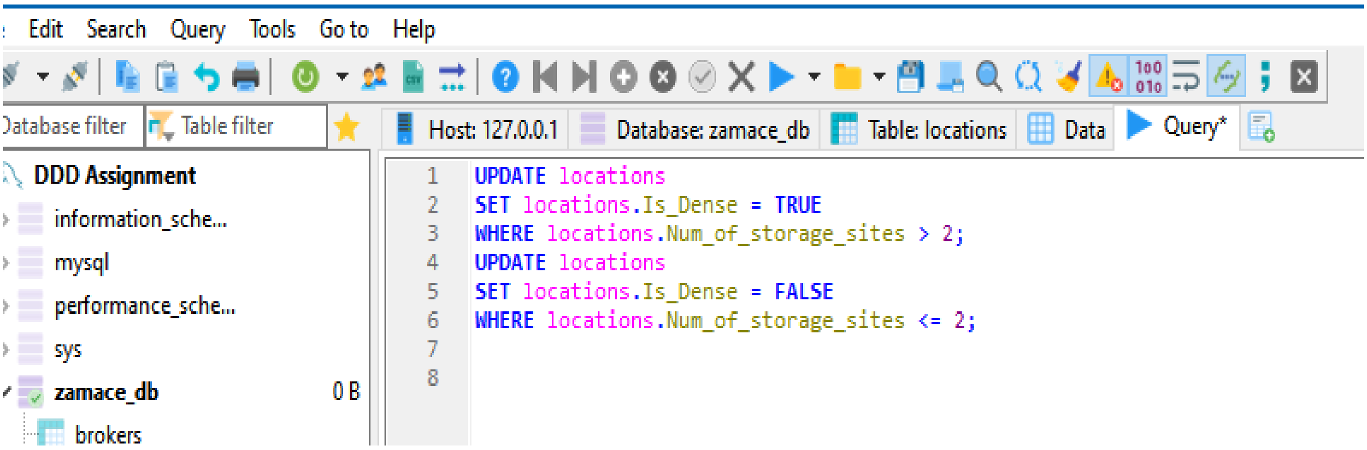


zamace_db.locations: 10 rows total (approximately)

Location_ID	Location_Name	Is_Dense	Circular_Area	Location_Capacity	Num_of_storage_sites
AAAA	Lusaka	(NULL)	1,000	100	4
AAAB	Mongu	(NULL)	700	70	2
AAAC	Livingstone	(NULL)	500	50	2
AAAD	Chongwe	(NULL)	700	70	3
AAAE	Mazabuka	(NULL)	564	56	2
AAAF	Chipata	(NULL)	444	44	1
AAAG	Lilongwe	(NULL)	176	17	2
AAAH	Kafue	(NULL)	232	23	3
AAAI	Kitwe	(NULL)	430	43	1
AAAJ	Ndola	(NULL)	545	54	1

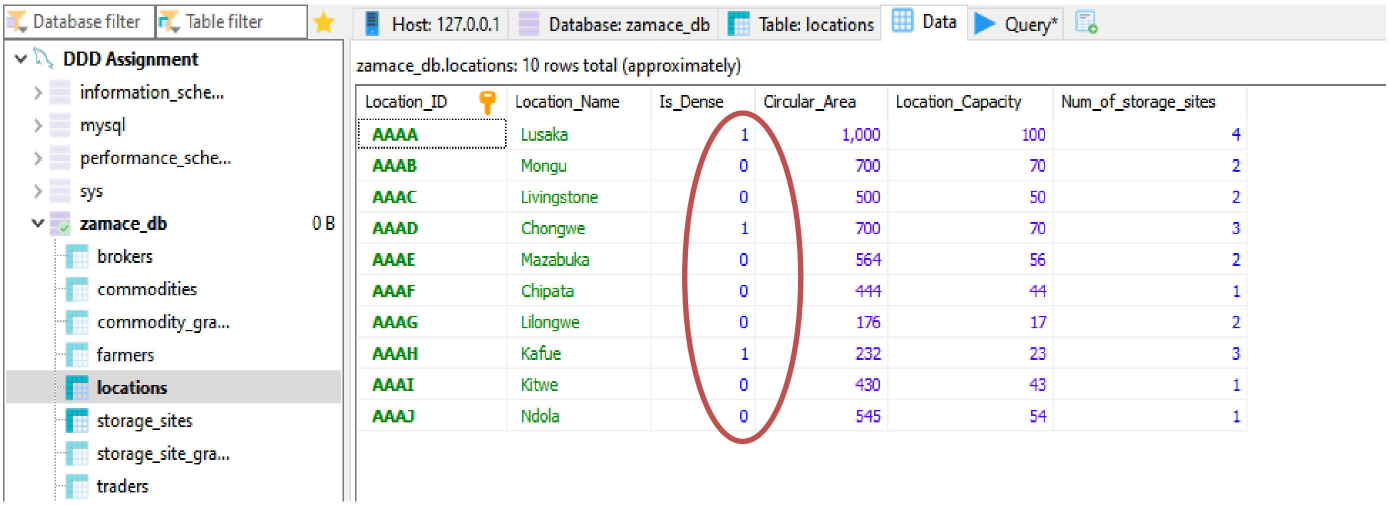
Query 8:

We want to also set the IS Dense column to TRUE if num of storage sites is greater than 2 otherwise FALSE.



```sql
UPDATE locations
SET locations.Is_Dense = TRUE
WHERE locations.Num_of_storage_sites > 2;
UPDATE locations
SET locations.Is_Dense = FALSE
WHERE locations.Num_of_storage_sites <= 2;
```

Results:



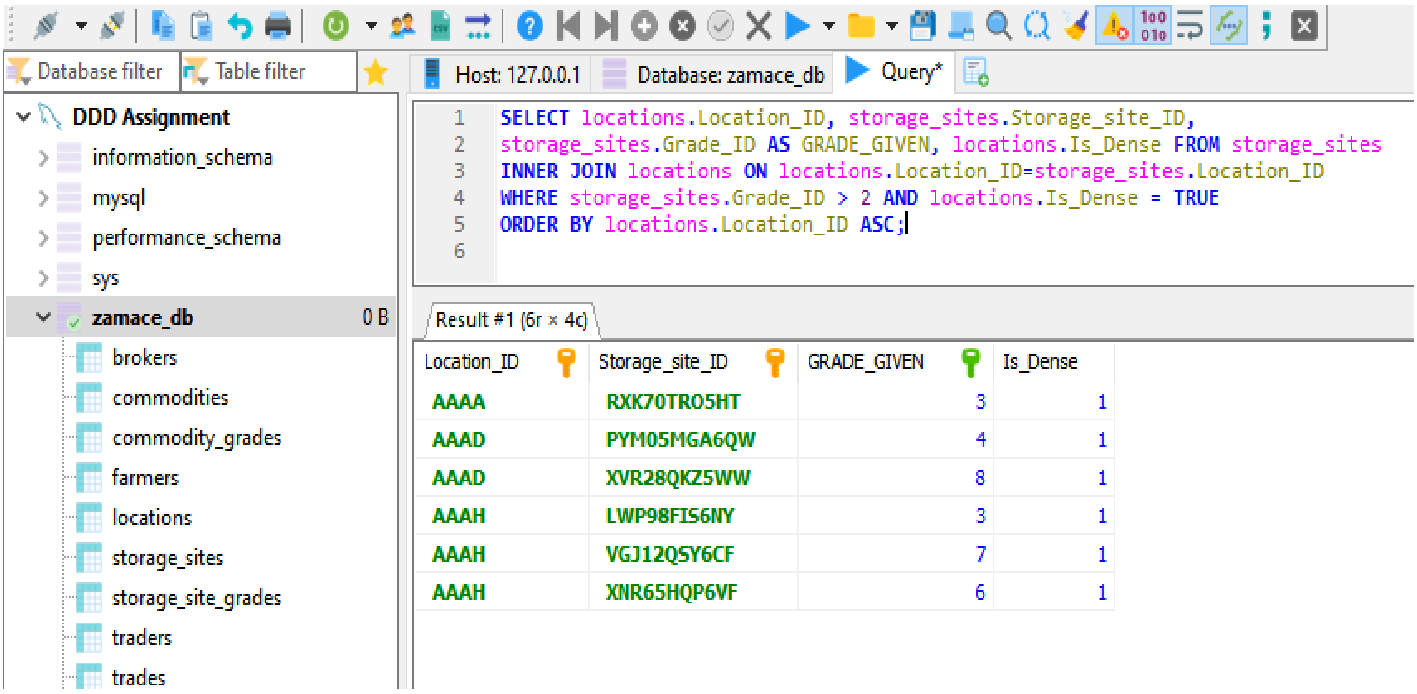
zamace_db.locations: 10 rows total (approximately)

Location_ID	Location_Name	Is_Dense	Circular_Area	Location_Capacity	Num_of_storage_sites
AAAA	Lusaka	1	1,000	100	4
AAAB	Mongu	0	700	70	2
AAAC	Livingstone	0	500	50	2
AAAD	Chongwe	1	700	70	3
AAAE	Mazabuka	0	564	56	2
AAAF	Chipata	0	444	44	1
AAAG	Lilongwe	0	176	17	2
AAAH	Kafue	1	232	23	3
AAAI	Kitwe	0	430	43	1
AAAJ	Ndola	0	545	54	1

TRUE evaluates to 1, FALSE evaluates to 0.

Query 9:

We want to list all storage sites in each location with the storage site grade which is above 2 and that location is dense.



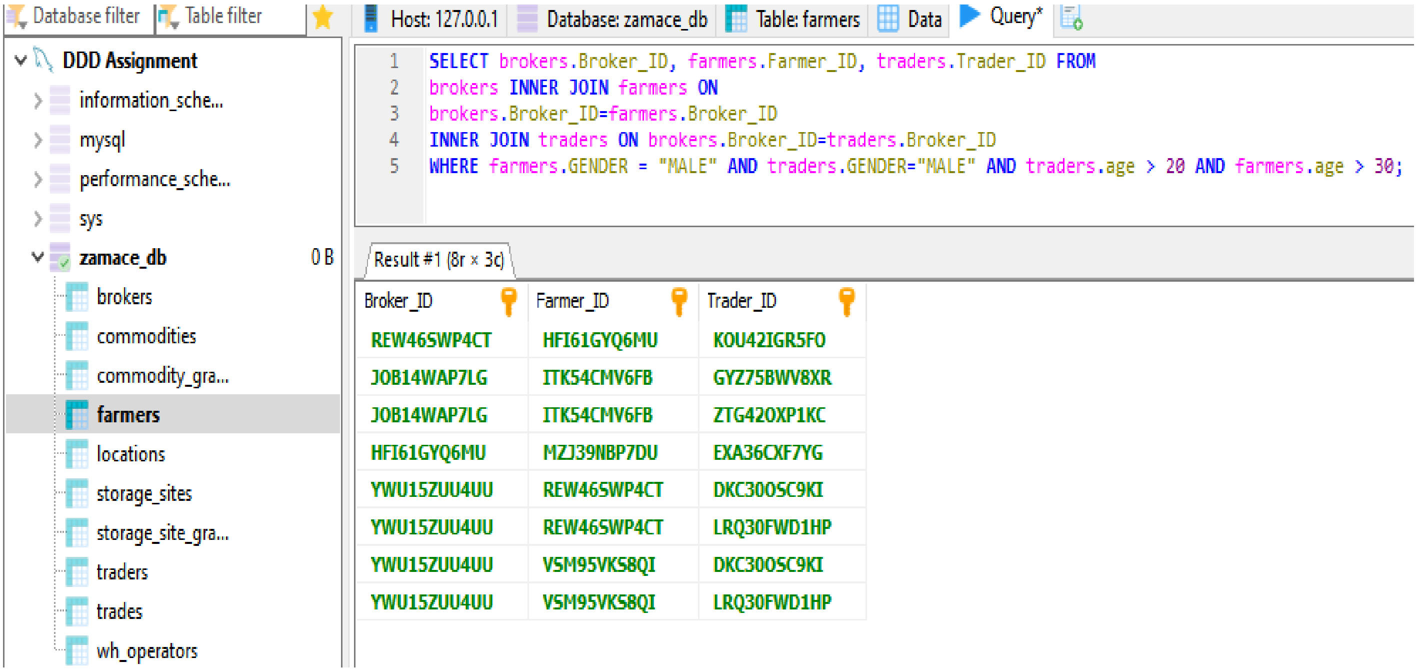
```sql
SELECT locations.Location_ID, storage_sites.Storage_site_ID,
storage_sites.Grade_ID AS GRADE_GIVEN, locations.Is_Dense FROM storage_sites
INNER JOIN locations ON locations.Location_ID=storage_sites.Location_ID
WHERE storage_sites.Grade_ID > 2 AND locations.Is_Dense = TRUE
ORDER BY locations.Location_ID ASC;
```

Result #1 (6r × 4c)

Location_ID	Storage_site_ID	GRADE_GIVEN	Is_Dense
AAAA	RXK70TRO5HT	3	1
AAAD	PYM05MGA6QW	4	1
AAAD	XVR28QKZ5WW	8	1
AAAH	LWP98FIS6NY	3	1
AAAH	VGJ12QSY6CF	7	1
AAAH	XNR65HQP6VF	6	1

Query 10:

We want to query only those brokers with their farmer clients and trader clients whose gender is male and above age 20.



```sql
SELECT brokers.Broker_ID, farmers.Farmer_ID, traders.Trader_ID FROM
brokers INNER JOIN farmers ON
brokers.Broker_ID=farmers.Broker_ID
INNER JOIN traders ON brokers.Broker_ID=traders.Broker_ID
WHERE farmers.GENDER = "MALE" AND traders.GENDER="MALE" AND traders.age > 20 AND farmers.age > 30;
```

Result #1 (8r × 3c)

Broker_ID	Farmer_ID	Trader_ID
REW46SWP4CT	HFI61GYQ6MU	KOU42IGR5FO
JOB14WAP7LG	ITK54CMV6FB	GYZ75BWV8XR
JOB14WAP7LG	ITK54CMV6FB	ZTG42OXP1KC
HFI61GYQ6MU	MZJ39NBP7DU	EXA36CXF7YG
YWU15ZUU4UU	REW46SWP4CT	DKC30OSC9KI
YWU15ZUU4UU	REW46SWP4CT	LRQ30FWD1HP
YWU15ZUU4UU	VSM95VKS8QI	DKC30OSC9KI
YWU15ZUU4UU	VSM95VKS8QI	LRQ30FWD1HP

TASK (9) (Data Warehouse)

A data warehouse, also known as an enterprise data warehouse, is a system used for data analysis and reporting in computing and is regarded as an essential part of business intelligence.

Factors:

We need a Data warehouse to keep historical records, even if the source transaction system does not. It will allow us to create and store metadata that helps data warehouse users understand data. Retrieving data from multiple data sources takes time;

therefore, in order to save time, we need to integrate the data from multiple sources and put it in a single location.

It prevents report users from misusing or corrupting transaction data; it can be used to store clean data that can be accessed by data analysts, businesses, team members, and data scientists; it will allow us to quickly run analysis on huge datasets; the complexity of queries will rise, and our users will require faster query processing; we will need a Data warehouse; users will be able to create their own reports without involving an IT specialist.

Common uses:

- Predicting customer churn based on sales data from the past ten years;
- dividing customers into distinct groups based on previous purchases to provide them with more detailed online content;
- and assisting businesses in the creation of demand and sales forecasts to determine which areas to focus on.

Our company will be able to develop efficient plans with the assistance of data warehouses if it is able to comprehend past data performances. It will provide our company with historical data, which we can use later to guide our business decisions. By utilizing a data warehouse, we will be able to precisely determine where our company currently stands and establish measurable branch marks that can drive long-term growth.

We will be able to check the most profitable commodities and warehouse operators in a certain amount of time in the past. Through this we can make useful business decisions to drive more profit to the company. We can find out the most profitable brokers to the company also and so on and so forth.

TASK (10) (Reflective Commentary)

Analysis and Description (what came and so what)

The assignment and course overall brought forward a variety of concepts about database design and development. This made me research and get a better understanding about how these concepts work. Concepts such as table constraints, propagation constraint's, the importance of normalization etc.

It is important to know these things because they save a whole lot of time and avoid confusion in designing a database for a company. I learnt how to create table with constraints, how to do join queries and how to alter tables. I have also noticed that I have a challenge in normalizing sometimes. I also found it difficult to follow some the SQL syntax because my DBMS had a different version. I also learnt that different DBMS can have different syntax sometimes to get a certain thing done.

I have also noticed that creating a database report is a challenge for me.

Future Plan (Now what)

I will learn more about data warehousing and put it into practice with a small-scale company by explaining to them how it works and its benefits.

I will practice more SQL queries and learn more about Triggers and more advanced concepts about SQL in the future.

I have to spend more time understanding the different variations of DBMS that support SQL i.e., MySQL, PostgreSQL, SQLite and so on.

I will learn how to integrate a database system into a Realtime application for real life use cases. A real time application such as a website, a bespoke software etc.

Finally, I will learn how to create more storage and performance efficient entities and performance efficient queries.

(REFERENCES)

Data warehousing and the factors that cause its need

https://aws.amazon.com/data-warehouse/#:~:text=A%20data%20warehouse%20is%20a,typically%20on%20a%20regular%20cadence.

Accessed on: 15/10/2022

Normalization and its importance

https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description#:~:text=Normalization%20is%20the%20process%20of,eliminating%20redundancy%20and%20inconsistent%20dependency.

Accessed on: 15/10/2022