Distribution Centre – Database Report

Database Description

When it comes to storing goods, we might think it is as simple as renting a building and turning it into a warehouse. But, the type of storage space we select can have an impact on our ability to satisfy the customers by fulfilling the orders on time. Therefore, it is very important to spend some time thinking which type of warehouse we need to support the business's objectives.

There are 6 different types of warehouse, distribution centre, climate-controlled warehouse, private warehouse, public warehouse, automated warehouse and fulfilment centre. In this project I have chosen to implement a database of a distribution centre.

Distribution warehouses are used to store and sell large quantities of goods. It typically houses goods from multiple manufactures/suppliers and sell them to the retailers. Goods are usually housed for short period of the time, sometimes just a day at a time. To run the supply chain smoothly they need to be able to serve a high volume of customers and house large amount of goods specially around the holiday seasons.

The database mainly describes the employees, equipment, suppliers, customers, delivery, storage and the products. I used staff, equipment, courier, customer, storage/shelf, product and supplier as a table to describe the warehouse. All the tables have an identifier called primary key. Some of the information from the tables references to another by a foreign key.

The staff table contains the staff_id, staff_name, date_of_birth, staff_address and job_id. Staff_id is the primary key and the foreign key is job_id which is referenced to storage_id in storage/shelf table.

The equipment table contains the equipment_id, equipment_name, equipment_type and user_id. Equipment_id is the primary key and the foreign key is user_id which is references to staff_id in staff table.

The supplier table contains supplier_id, supplier_name, delivery_item, phone_num and customer_id. Supplier_id is the primary key and the foreign key is customer_id which is references to customer_id in customer table.

The courier table contains the delivery_id, destination, dept_time and arrivial_time. Delivery_time is the primary key of the table.

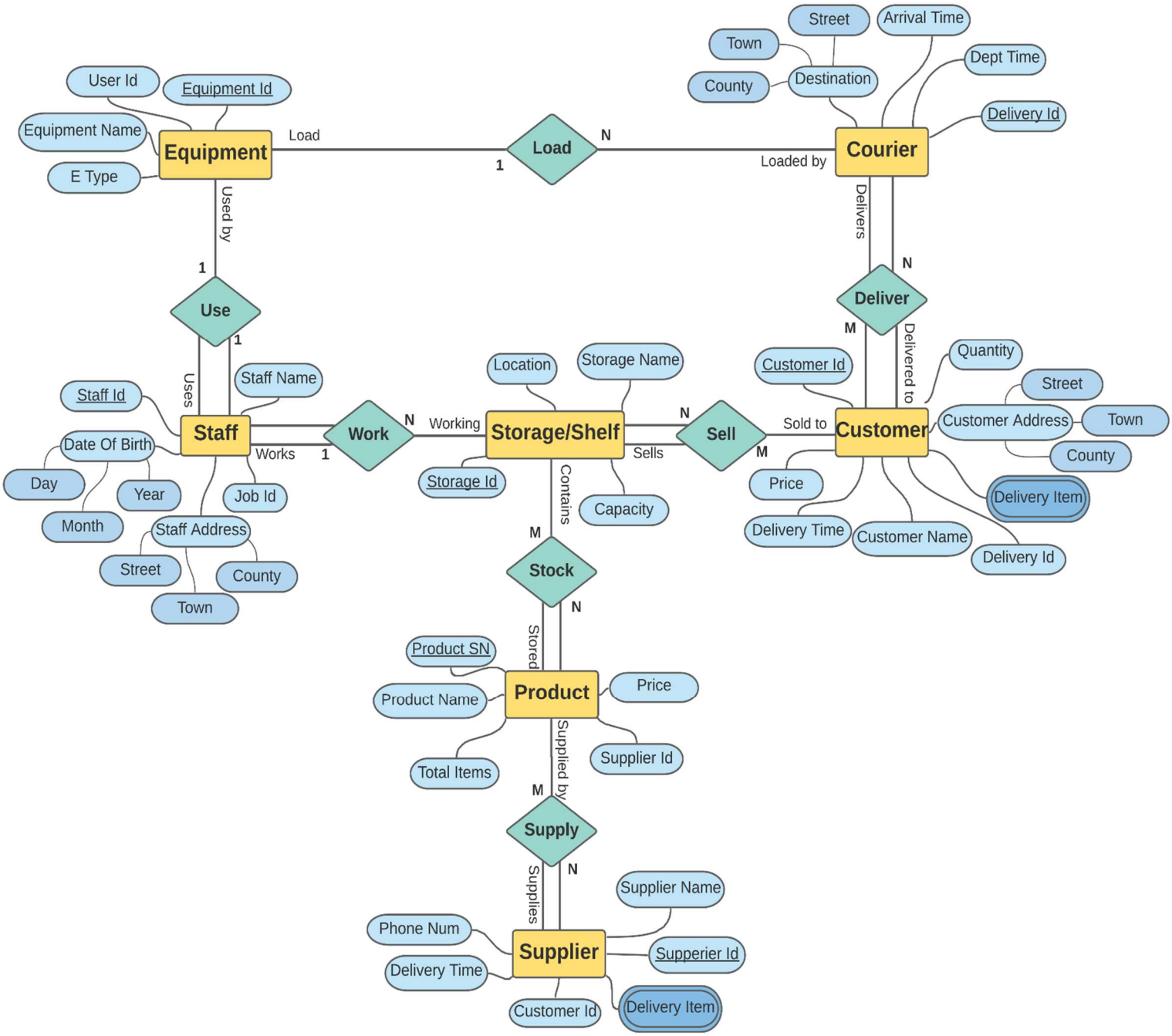
The customer table contains the customer_id, customer_name, customer_address, delivery_item, delivery_id, delivery_time, price and quantity. Customer_id is the primary key of the table.

The storage/shelf table contains the storage_id, location, storage_name and the capacity. Storage_id is the primary key of the table.

The product table contains the product_sn, product_name, total_items, price and supplier. Product_sn is the primary key and the foreign key is supplier_id which references to supplier_id in supplier table.

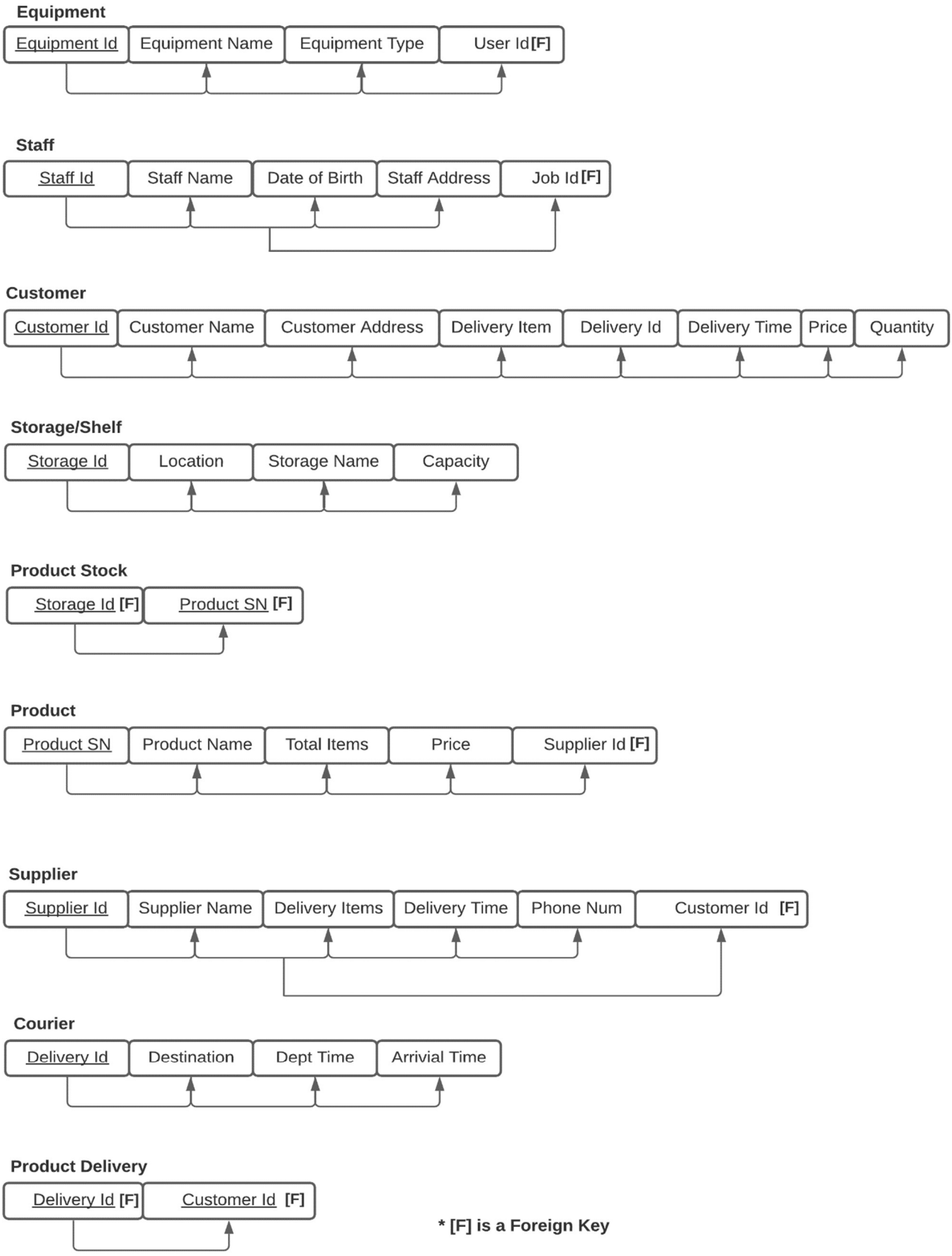
Entity Relationship Diagram

The entity relationship diagram for the database is shown below:



Functional Dependency Diagram

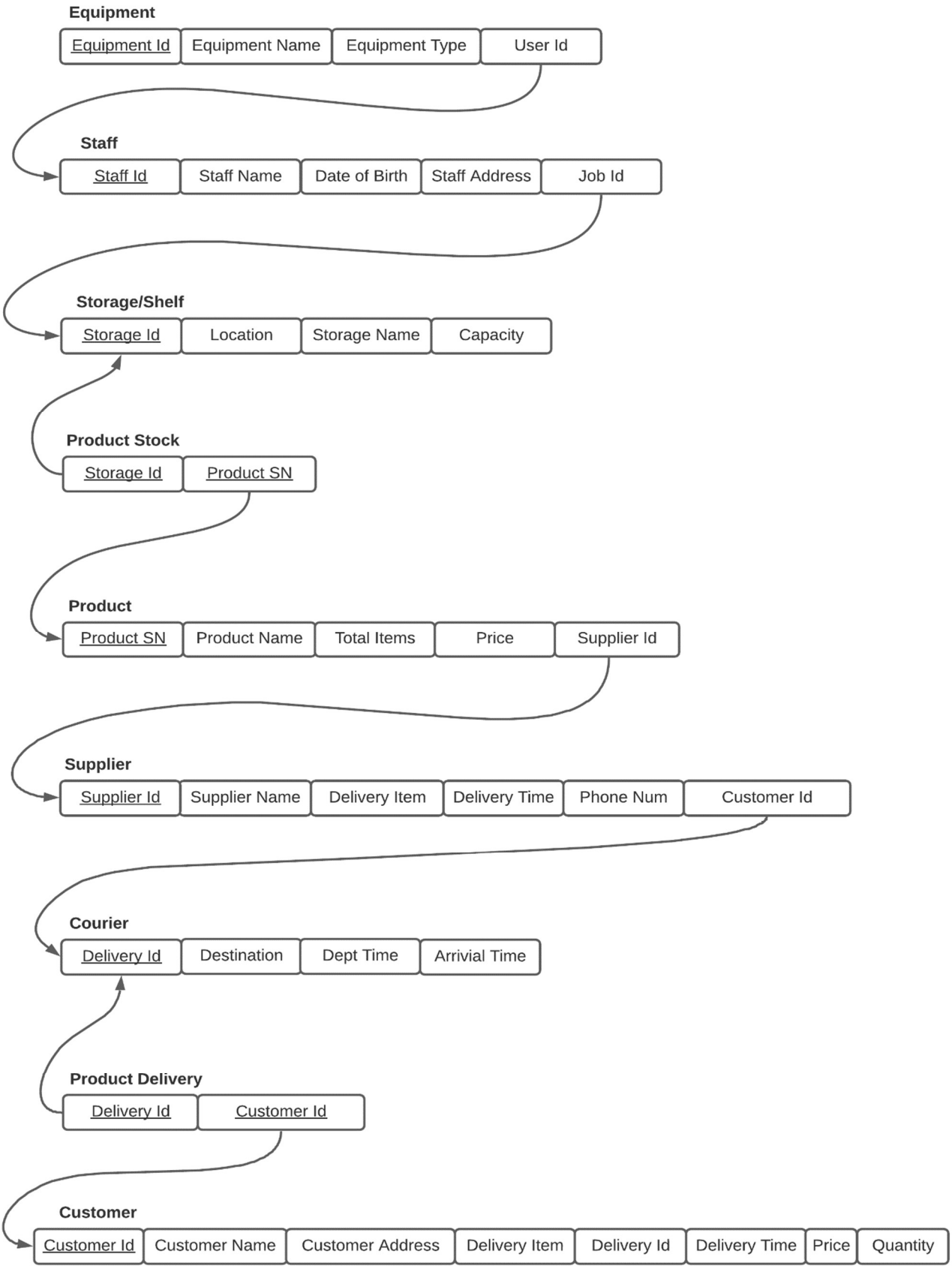
The functional dependency diagram of the database is shown below:



*** [F] is a Foreign Key**

Mapping to Relational Schema

The mapping from the entity relationship diagram to an outline relational schema is shown below:



Implicit Constraints

Equipment

Primary Key:

Equipment_Id VARCHAR (10) not null

Foreign Key(s):

Foreign Key (User_Id) References Staff (Staff_Id)

Staff

Primary Key:

Staff_Id Integer not null

Foreign Key(s):

Foreign Key (Job_Id) References Shelf (Storage_Id)

Storage/Shelf

Primary Key:

Storage_Id Varchar (20) not null

Foreign Key(s):

Null

Product

Primary Key:

Product_SN INTEGER not null

Foreign Key(s):

Foreign Key (Supplier_Id) References Supplier (Supplier_Id),

Supplier

Primary Key:

Supplier_Id INTEGER not null

Foreign Key(s):

Foreign Key (Customer_Id) References Courier (Delivery_Id),

Courier

Primary Key:

Delivery_Id Integer not null

Foreign Key(s):

Null

Customer

Primary Key:

Customer_Id Integer not null

Foreign Key(s):

Null

Semantic Constraints

There are many semantic constraints in this database. Very first constraint is in the customer identification number, it can only be 8 digit long.

```
CONSTRAINT Check_Customer_Id CHECK(LENGTH(Customer_Id) = 8)
```

There is also a constraint on staff's job_id. It checks if the job_id given to the staff exists or not.

```
CONSTRAINT Check_Job_Id CHECK (Job_Id IN ('Elec0001', 'OutAcc0501', 'MuscDJ0007', 'HFHA0001', 'FaClAc01', 'HeBe001', 'GaToy01', 'HoFur123')
```

Triggers

One of the triggers occurs when a employee leaves the company and new person is hired. New employee will continue in the same position as the last employee.

```
CREATE OR REPLACE TRIGGER remove_staff_update_equipment
AFTER DELETE ON Staff
FOR EACH ROW
WHEN (OLD.Staff_Id IS NOT NULL)
DECLARE
        Del_Staff_Id NUMBER;
BEGIN
        Del_Staff_Id := : OLD.Staff_Id;
        UPDATE Equipment
        SET User_Id = NULL
        WHERE User_Id = Del_Staff_Id;
END remove_staff_update_equipment;
.
RUN;
```

The trigger below occurs when a new employee is hired in the warehouse. His/her id, name, date of birth, address and job id will be inserted to the database.

```
CREATE OR REPLACE TRIGGER Add_Staff
AFTER INSERT ON Staff
FOR EACH ROW
DECLARE
        Add_Staff_Id NUMBER;
BEGIN
        Add_Staff_Id := :NEW.Staff_Id;
        UPDATE Equipment
        SET User_Id = User_Id + 1
        WHERE User_Id = Add_Staff_Id;
END Add_Staff;
.
RUN;
```

Update Operations

There are many update operations which can be performed on this database. Below are some of the commands we can use to update the information on the database.

1. If the employee 0123456 wants to move from Electronics & Office to Musical Instruments & DJ, the information can be updated as following:

```
UPDATE Staff SET Job_Id = 'MuscDJ0007' WHERE Staff_Id = 0123456;
```

2. If the capacity of the Storage HoFur123(Home and Furniture) is increased from 700 to 1000, the information can be updated as following:

```
UPDATE Shelf SET Capacity = 1000 WHERE Storage_Id = 'HoFur123';
```

Relational Select Operations

Select operation on a database can be performed as shown below:

1. Select all the Staffs in the warehouse:

```
SELECT * FROM Staff;
```

Staff_Id	Staff_Name	Date_of_Birth	Staff_Address	Job_Id
123456	Roxy Tollemache	10/02/1990	92 Water Rd, Saint Johns, South Dublin	MuscDJ0007
123457	Terry Brine	16/10/1989	66 High St, Saint Marks, Kilkenny	OutAcc0501
123458	Michael Kennedy	21/04/1995	12 Kill Lane, View park, Waterford	MuscDJ0007
123459	Peter Doyle	27/02/1986	47 Kenyon St, Nenagh, Dublin	HFHA0001
123460	Alan McCarthy	02/02/1998	Duffy Hill, Enniscorthy, South Dublin	FaClAc01
123461	Conor Doyle	07/03/1981	104 Creagh Ct., Rathcoole, Dublin	HeBe001
123462	Mark Walsh	16/06/1976	The Square, Oldcastle, South Dublin	GaToy01
123463	Dave Smith	19/10/1987	10 Hanover Qy., Tel, Clondalkin, Dublin	HoFur123

2. Select all the products that have over 500 items:

```
SELECT*FROM Product WHERE Total_Items > 500;
```

Product_SN	Product_Name	Total_Items	Price	Supplier_Id
1111	Sony Headphones	1000	€219,000	123456780
1112	Amazon Echo Dot	1000	€29,990	123456791
1113	Tommy Hilfiger Backpack	1000	€64,540	123456802

Security

There are many concerns around security of a database. Misuse and unauthorised activities are some of the main issues. To maintain the security of a database there is limited access of the information to the staffs. Only the senior and the manager can remove or update the information. For example, staffs can see who works in which department, but they will not have access to other staff's home address or date of birth etc. The main objective of this is to hide any sensitive information of the employees.

Appendix A – Table Creation

```
CREATE DATABASE warehouse;
USE warehouse;
CREATE TABLE Customer (
        Customer_Id Integer not null,
        Customer_Name Varchar (20) not null,
        Customer_Address Varchar (50) not null,
        Delivery_item Varchar (50) not null,
        Delivery_Id Integer not null,
        Delivery_Time Varchar (10) not null,
        Quantity Integer not null,
        Price Varchar (20) not null,
        Primary Key (Customer_Id),
        CONSTRAINT Check_Customer_Id CHECK(LENGTH(Customer_Id) = 8),
        CONSTRAINT Check_Cus_Delivery_Id CHECK(LENGTH(Delivery_Id) = 8)
        );
ALTER TABLE Customer
ADD CONSTRAIN Min_Quantity CHECK (Quantity > 0);
ALTER TABLE Customer add unique (Customer_Id);

CREATE TABLE Courier (
        Delivery_Id Integer not null,
        Destination Varchar (50) not null,
        Dept_Time Varchar (10) not null,
        Arrival_Time Varchar (20) not null,
        Primary Key (Delivery_Id),
        CONSTRAINT Check_Delivery_Id CHECK(LENGTH(Delivery_Id) = 8)
        );
ALTER TABLE Courier add unique (Delivery_Id);
```

```
CREATE TABLE Shelf (
        Storage_Id Varchar (20) not null,
        Location Varchar (20) not null,
        Storage_Name Varchar (50) not null,
        Capacity Varchar (20) not null,
        Primary Key (Storage_Id),
        CONSTRAINT Quantity_Check CHECK (LENGTH (Quantity) <= 5000)
       );

CREATE TABLE Staff (
        Staff_Id Integer not null,
        Staff_Name Varchar (20) not null,
        Date_of_Birth Varchar (10) not null,
        Staff_Address Varchar (50) not null,
        Job_Id Varchar (20) not null,
        Primary Key (Staff_Id),
        Foreign Key (Job_Id) References Shelf (Storage_Id),
        CONSTRAINT Check_Job_Id CHECK (Job_Id IN ('Elec0001', 'OutAcc0501', 'MuscDJ0007',
'HFHA0001', 'FaClAc01', 'HeBe001', 'GaToy01', 'HoFur123'))
       );
ALTER TABLE Staff add unique (Staff_Id);

CREATE TABLE Equipment (
        Equipment_Id VARCHAR (10) not null,
        Equipment_Name VARCHAR (50) not null,
        Equipment_Type VARCHAR (50) not null,
        User_Id INTEGER not null,
        Primary Key (Equipment_Id),
        Foreign Key (User_Id) References Staff (Staff_Id)
       );
```

```
CREATE TABLE Supplier (
        Supplier_Id INTEGER not null,
        Supplier_Name VARCHAR (50) not null,
        Delivery_item Varchar (50) not null,
        Customer_Id Integer not null,
        Delivery_Time Varchar (10) not null,
        Phone_Num Varchar (20) not null,
        Primary Key (Supplier_Id),
        Foreign Key (Customer_Id) References Courier (Delivery_Id),
        CONSTRAINT Check_Supplier_Id CHECK(LENGTH(Supplier_Id) = 9)
        );

CREATE TABLE Product (
        Product_SN INTEGER not null,
        Product_Name VARCHAR (50) not null,
        Total_Items INTEGER not null,
        Price VARCHAR (10) not null,
        Supplier_Id INTEGER not null,
        Primary Key (Product_SN),
        Foreign Key (Supplier_Id) References Supplier (Supplier_Id),
        CONSTRAINT Check_Total_Items CHECK(LENGTH(Total_Items) > 0)
        );

CREATE TABLE Product_Stock (
        Storage_Id Varchar (20) not null,
        Product_SN Integer not null,
        Primary Key (Storage_Id, Product_SN),
        Foreign Key (Storage_Id) References Shelf (Storage_Id),
        Foreign Key (Product_SN) References Product (Product_SN)
        );
```

```
CREATE TABLE Product_Delivery (
        Delivery_Id INTEGER not null,
        Customer_Id Integer not null,
        Primary Key (Delivery_Id, Customer_Id),
        Foreign Key (Delivery_Id) References Courier (Delivery_Id),
        Foreign Key (Customer_Id) References Customer (Customer_Id)
       );
```

Appendix B – Database Population

```
INSERT INTO Customer VALUES (12345678, 'PC Universe', '1 Parkview, Greenlanes, Dublin 3', 'Sony Headphones', 87654321, '12:30', 50, '€10,950.00');

INSERT INTO Customer VALUES (12345679, 'Cool Tech', '2 Seaview, Maryland, Dublin 5', 'Amazon Echo Dot', 87654322, '18:00', 30, '€899.70');

INSERT INTO Customer VALUES (12345680, 'Bag Life', '12 Moonwalk, Waterview, Dublin 1', 'Tommy Hilfiger Backpack', 87654323, '15:30', 20, '€1,290.80');

INSERT INTO Customer VALUES (12345681, 'Mulls Music', '53 Sideview, Chelsea Gardens, Dublin 2', 'Wooden Hand Drum', 87654324, '13:30', 15, '€346.35');

INSERT INTO Customer VALUES (12345682, 'Techo', '22 Northsides Park, Swords, Dublin', 'Elgato Stream Deck', 87654325, '11:30', 5, '€1,049.95');


INSERT INTO Courier VALUES (23456789, '1 Parkview, Greenlanes, Dublin 3', '06:00', '12:30');

INSERT INTO Courier VALUES (23456790, '2 Seaview, Maryland, Dublin 5', '06:30', '18:00');

INSERT INTO Courier VALUES (23456791, '12 Moonwalk, Waterview, Dublin 1', '07:30', '15:30');

INSERT INTO Courier VALUES (23456792, '53 Sideview, Chelsea Gardens, Dublin 2', '07:00', '13:30');

INSERT INTO Courier VALUES (23456793, '22 Northsides Park, Swords, Dublin', '08:00', '11:30');


INSERT INTO Shelf VALUES ('Elec0001', '2nd Level', 'Electronics & Office', 5000);

INSERT INTO Shelf VALUES ('Elec0002', '2nd Level', 'Electronics & Office', 1000);

INSERT INTO Shelf VALUES ('OutAcc0501', '1st Level', 'Outdoors & Accessories', 2500);

INSERT INTO Shelf VALUES ('MuscDJ0007', '3rd Level', 'Musical Instruments & DJ', 500);

INSERT INTO Shelf VALUES ('HFHA0001', '4th Level', 'Hi-Fi & Home Audio', 1000);

INSERT INTO Shelf VALUES ('FaClAc01', '3rd Level', 'Fashion, Clothing & Accessories', 5000);

INSERT INTO Shelf VALUES ('HeBe001', '3rd Level', 'Health & Beauty', 5000);

INSERT INTO Shelf VALUES ('GaToy01', '1st Level', 'Gaming & Toys', 4500);

INSERT INTO Shelf VALUES ('HoFur123', '0 Level', 'Home and Furniture', 700);


INSERT INTO Staff VALUES (0123456, 'Roxy Tollemache', '10/02/1990', '92 Water Rd, Saint Johns, South Dublin', 'Elec0001');

INSERT INTO Staff VALUES (0123457, 'Terry Brine', '16/10/1989', '66 High St, Saint Marks, Kilkenny', 'OutAcc0501');

INSERT INTO Staff VALUES (0123458, 'Michael Kennedy', '21/04/1995', '12 Kill Lane, View park, Waterford', 'MuscDJ0007');
```

```
INSERT INTO Staff VALUES (0123459, 'Peter Doyle', '27/02/1986', '47 Kenyon St, Nenagh, Dublin', 'HFHA0001');

INSERT INTO Staff VALUES (0123460, 'Alan McCarthy', '02/02/1998', 'Duffy Hill, Enniscorthy, South Dublin', 'FaClAc01');

INSERT INTO Staff VALUES (0123461, 'Conor Doyle', '07/03/1981', '104 Creagh Ct., Rathcoole, Dublin', 'HeBe001');

INSERT INTO Staff VALUES (0123462, 'Mark Walsh', '16/06/1976', 'The Square, Oldcastle, South Dublin', 'GaToy01');

INSERT INTO Staff VALUES (0123463, 'Dave Smith', '19/10/1987', '10 Hanover Qy., Tel, Clondalkin, Dublin', 'HoFur123');


INSERT INTO Equipment VALUES ('HPT001', 'Hand pallet truck', 'Manual', 0123456);

INSERT INTO Equipment VALUES ('SEPT001', 'Semi-electric pallet truck', 'Semi-electric', 0123462);

INSERT INTO Equipment VALUES ('EPT001', 'Electric pallet truck', 'Electric', 0123459);

INSERT INTO Equipment VALUES ('MS001', 'Manual stacker', 'Manual', 0123461);

INSERT INTO Equipment VALUES ('SES001', 'Semi-electric stacker', 'Semi-electric', 0123457);

INSERT INTO Equipment VALUES ('ES001', 'Electric stacker', 'Electric', 0123463);


INSERT INTO Supplier VALUES (123456780, 'Sony Store', 'Sony Headphones', 23456789, '13:30', '+353876543210');

INSERT INTO Supplier VALUES (123456791, 'Amazon', 'Amazon Echo Dot', 23456790, '12:30', '+353876543211');

INSERT INTO Supplier VALUES (123456802, 'Tommy Hilfiger', 'Tommy Hilfiger Backpack', 23456791, '11:00', '+353876543212');

INSERT INTO Supplier VALUES (123456813, 'Perfeclan', 'Wooden Hand Drum', 23456792, '15:30', '+353876543213');

INSERT INTO Supplier VALUES (123456824, 'Elgato Store', 'Elgato Stream Deck', 23456793, '17:00', '+353876543214');


INSERT INTO Product VALUES (00001111, 'Sony Headphones', 1000, '€219,000', 123456780);

INSERT INTO Product VALUES (00001112, 'Amazon Echo Dot', 1000, '€29,990', 123456791);

INSERT INTO Product VALUES (00001113, 'Tommy Hilfiger Backpack', 1000, '€64,540', 123456802);

INSERT INTO Product VALUES (00001114, 'Wooden Hand Drum', 500, '€11,545', 123456813);

INSERT INTO Product VALUES (00001115, 'Elgato Stream Deck', 500, '€104,995', 123456824);
```

```
INSERT INTO Product_Stock VALUES ('Elec0001', 00001111);
INSERT INTO Product_Stock VALUES ('Elec0002', 00001112);
INSERT INTO Product_Stock VALUES ('OutAcc0501', 00001113);
INSERT INTO Product_Stock VALUES ('MuscDJ0007', 00001114);
INSERT INTO Product_Stock VALUES ('HFHA0001', 00001115);

INSERT INTO Product_Delivery VALUES (23456789, 12345678);
INSERT INTO Product_Delivery VALUES (23456790, 12345679);
INSERT INTO Product_Delivery VALUES (23456791, 12345680);
INSERT INTO Product_Delivery VALUES (23456792, 12345681);
INSERT INTO Product_Delivery VALUES (23456793, 12345682);
```