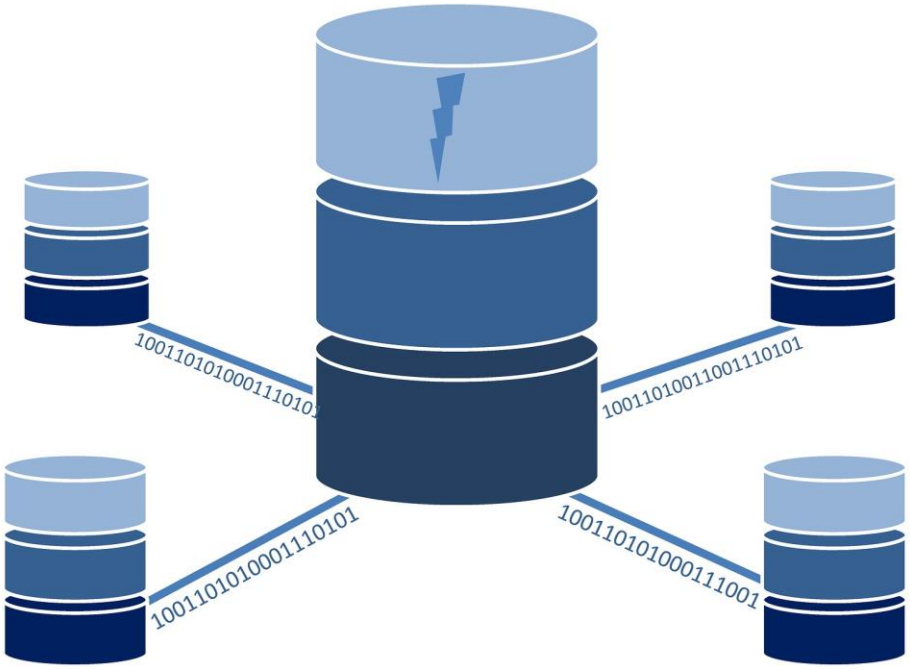
Warehouse Database



DATABASE SYSTEMS (1) CSE227
SUBMITTED TO: DR. HODA KORASHY

Mohamed Sameh Abdelhakim 16p3061

Nour E-din Osama Mohamed 16p6043

Youssef Khaled Hussein 16p6040

Contents

Description

This project's aim is to build a database system for a warehouse specialized in storing tech merchandise (TVs, Mobiles, Games, etc.) from the point they are received from a supplier till they are handed off to the proper distributor.

The database shall be responsible for keeping track of the complete life span of items inside the warehouse including all the data. However, this database will not include data like staff data, operating hours, etc. which can be added easily if needed.

Requirements

ENTITIES

Initial conversation with the stakeholders resulted in the following entities:

- Item: representing any singular item in the database, meaning that if we have 2 identical phones in the warehouse they will be saved in the database as 2 entries because it is required to have information about every single item inside the warehouse to be able to ship different amounts to different distributors.
  For each entity we need to keep its type, ID, time of arrival, time or shipping, location, color, size, description, producing company, screen size for TVs and Mobiles, Genre and platform for video games and voltage for chargers. It should be noticed that some of those attributed like size will be common among all identical items in the warehouse.
- Supplier: represent the company that supplies items to the warehouse. Every item has a supplier, but a supplier may not have supplied current items at a time.
  For each Supplier we need to keep their ID, contract start and termination dates, identifying information like mail, phone, address, etc. A supplier must be a company not an individual or a store.
- Distributor: The entity that receives items from the warehouse.
  For each Distributor we need to keep their ID, Tax register number, some identifying info like in the supplier as well as SSN for individual suppliers and commerce register number for companies.

DERIVED ENTITIES

The following entities were derived based on the initial requirement to represent conflicts in attribute under the same entity:

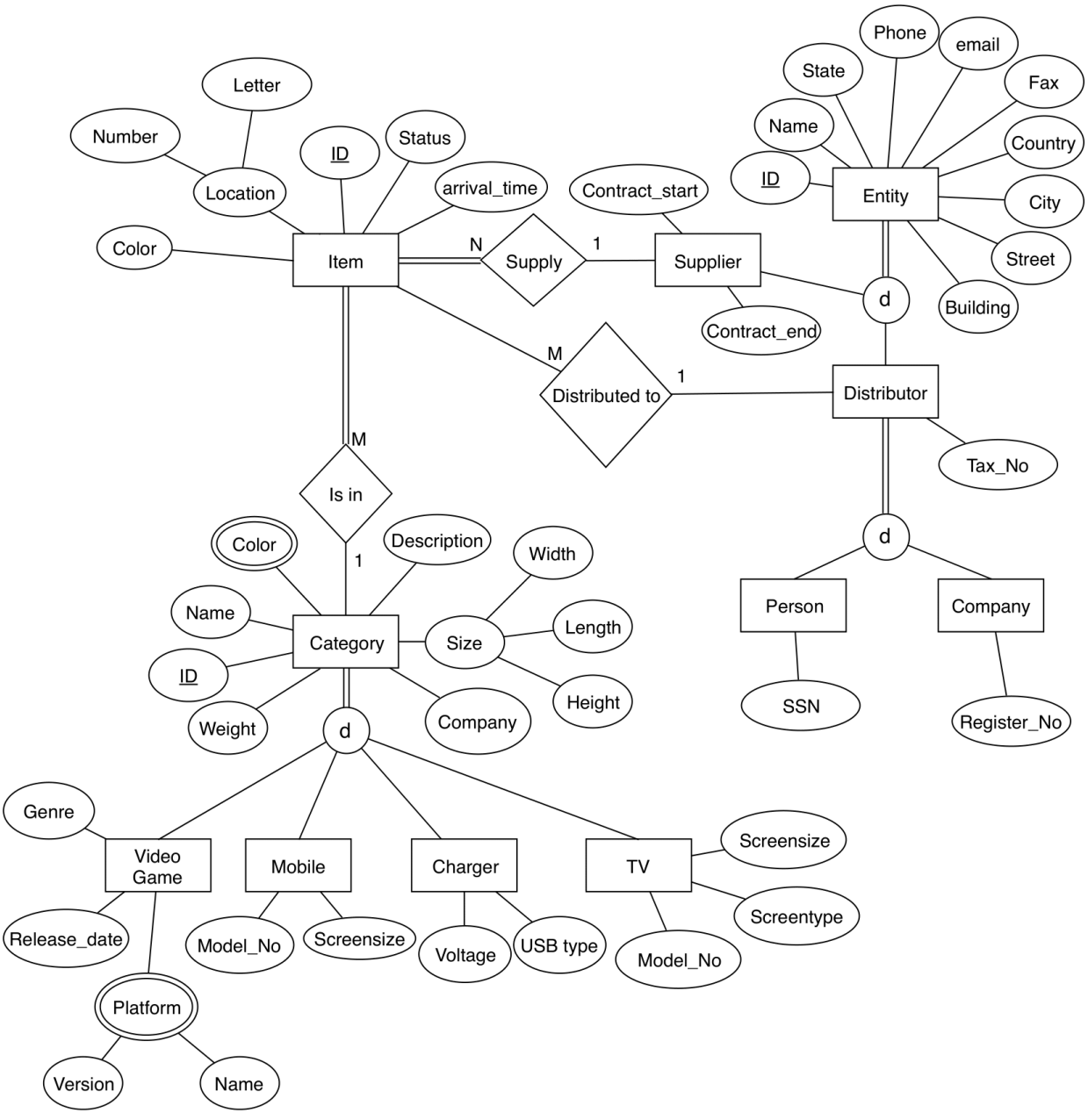
- Category: represents the model of an item to keep the common attributes across multiple items like weight, size, available colors, description and Name.
- Video Game: a category that holds platform information and genre.
- Mobile: a category that holds screen size and model number.

- Charger: a category that holds output voltage.
- TV: a category that holds Model no. and screen type .
- Person: a distributor representing a person with an SSN.
- Company: a distributor representing a company with a commercial register number.
- Entity: a superclass for distributor and supplier to hold their common information like email, fax, address, Name, etc.

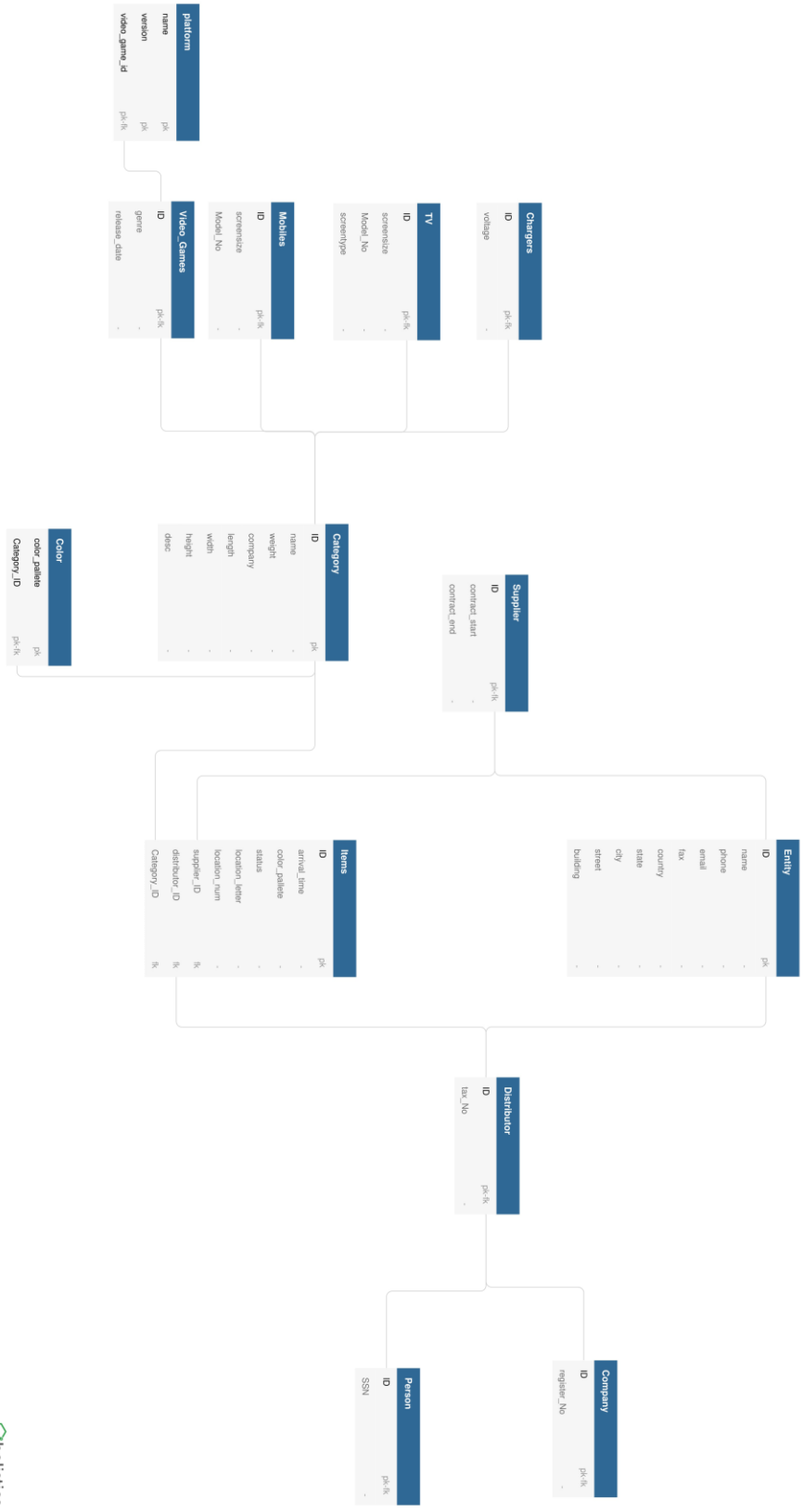
Assumptions

1. Item must be supplied by one supplier only while a supplier may supply many items
2. Distributor may receive many items, item may be sent to multiple distributors
3. Item must have one category, but a category may have many items.
4. platform is a multivalued composite attribute in videogame
5. A distributor is either a company or a person
6. Both distributors and suppliers have a common superclass called entity
7. Color is a multivalued attribute in category as one category can have multiple possible colors; however, an item has one color only.
8. Location is a composite attribute in item consisting of a letter and a number(E.g. A5).
9. TIMESTAMP is used for attributes associated with time.
10. INTEGER is used for all IDs and any value that is an integer in nature
11. FLOAT is used for size, voltage and screen size.
12. VARCHAR and CHAR are used for all other generic attributes like SSN.

EER Diagram


Letter
Number
Location
ID
Status
arrival_time
Color
Item
N
Supply
1
Contract_start
Supplier
Contract_end
Phone
email
State
Fax
Name
Country
ID
Entity
City
Street
Building
d
M
Distributed to
1
Distributor
Tax_No
M
Is in
1
Color
Description
Width
Name
Category
Size
Length
ID
Height
Weight
Company
d
Person
Company
SSN
Register_No
Genre
Video Game
Mobile
Charger
TV
Screensize
Screentype
Release_date
Model_No
Screensize
Voltage
USB type
Model_No
Platform
Version
Name

Relational Model



SQL Sample

CREATING TABLES

```sql
CREATE TABLE Entity
(
    ID INTEGER PRIMARY KEY NOT NULL,
    name VARCHAR(100) NOT NULL,
    phone VARCHAR(20) NOT NULL,
    email VARCHAR(25) NOT NULL,
    fax VARCHAR(30),
    country VARCHAR(20) NOT NULL,
    state VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,
    street VARCHAR(20) NOT NULL,
    building VARCHAR(20) NOT NULL
);

CREATE TABLE Supplier
(
    ID INTEGER PRIMARY KEY NOT NULL,
    contract_start DATE NOT NULL,
    contract_end DATE,
    FOREIGN KEY (ID) REFERENCES Entity (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Distributor
(
    ID INTEGER PRIMARY KEY NOT NULL,
    tax_No VARCHAR(40) NOT NULL,
    FOREIGN KEY (ID) REFERENCES Entity (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Company
(
    ID INTEGER PRIMARY KEY NOT NULL,
    register_No VARCHAR(40) NOT NULL,
    FOREIGN KEY (ID) REFERENCES Distributor (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Person
(
    ID INTEGER PRIMARY KEY NOT NULL,
    SSN VARCHAR(20) NOT NULL,
    FOREIGN KEY (ID) REFERENCES Distributor (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Items
(
    ID INTEGER PRIMARY KEY NOT NULL,
    arrival_time TIMESTAMP NOT NULL,
```

```
    color_pallete VARCHAR(10) NOT NULL,
    `status` varchar(255) NOT NULL,
    location_letter CHAR(1) NOT NULL,
    location_num CHAR(3) NOT NULL,
    supplier_ID INTEGER NOT NULL,
    distributor_ID INTEGER,
    Category_ID INTEGER NOT NULL,
    FOREIGN KEY (supplier_ID) REFERENCES Supplier (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
    FOREIGN KEY (distributor_ID) REFERENCES Distributor(ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
    FOREIGN KEY (Category_ID) REFERENCES Category (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Category
(
    ID INTEGER PRIMARY KEY NOT NULL,
    name VARCHAR(100) NOT NULL,
    weight FLOAT NOT NULL,
    company VARCHAR(50) NOT NULL,
    length FLOAT NOT NULL,
    width FLOAT NOT NULL,
    height FLOAT NOT NULL,
    `desc` VARCHAR(500)
);

CREATE TABLE TV
(
    ID INTEGER PRIMARY KEY NOT NULL,
    screensize FLOAT NOT NULL,
    Model_No VARCHAR(20),
    screentype VARCHAR(10),
    FOREIGN KEY (ID) REFERENCES Category (ID)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Chargers
(
    ID INTEGER PRIMARY KEY NOT NULL,
        voltage FLOAT NOT NULL,
        charger_Type VARCHAR(40) NOT NULL,
        FOREIGN KEY (ID) REFERENCES Category (ID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
    );

    CREATE TABLE Mobiles
    (
        ID INTEGER PRIMARY KEY NOT NULL,
        screensize FLOAT NOT NULL,
        Model_No VARCHAR(20) NOT NULL,
        FOREIGN KEY (ID) REFERENCES Category (ID)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
    );

```

```
117.        CREATE TABLE Video_Games
118.        (
119.            ID INTEGER PRIMARY KEY NOT NULL,
120.            genre VARCHAR(20),
121.            release_date DATE,
122.            FOREIGN KEY (ID) REFERENCES Category (ID)
123.            ON DELETE RESTRICT
124.            ON UPDATE CASCADE
125.        );
126.
127.        CREATE TABLE platform
128.        (
129.            name VARCHAR(20) NOT NULL,
130.            version VARCHAR(20) NOT NULL,
131.            video_game_id INTEGER NOT NULL,
132.            PRIMARY KEY (name,version,video_game_id),
133.            FOREIGN KEY (video_game_id) REFERENCES Video_Games (ID)
134.            ON DELETE RESTRICT
135.            ON UPDATE CASCADE
136.        );
137.
138.        CREATE TABLE Color
139.        (
140.            color_pallete VARCHAR(10) NOT NULL,
141.            Category_ID INTEGER NOT NULL,
142.            PRIMARY KEY (color_pallete, category_ID),
143.            FOREIGN KEY (Category_ID) REFERENCES Category (ID)
144.            ON DELETE RESTRICT
145.            ON UPDATE CASCADE
146.        );
```

INSERTING INTO TABLES

```
1.  INSERT INTO Entity
2.  Values (1, "Westroos Company",+20111, "amidn@westros.com",+20112,"Egypt","Cairo"
    ,"Mokattam","nine","8127");
3.
4.  INSERT INTO Supplier
5.  Values (1,'2019-5-8','2019-12-30');
6.
7.
8.
9.  INSERT INTO Entity
10. Values (2, "winterfell Company",+20111, "amidn@winterfell.com",+20223,"Spain","b
    arcelona","NAN","81","54");
11.
12. INSERT INTO Distributor
13. Values (2,"99987887");
14.
15. INSERT INTO Company
16. values (2,"555555");
17.
18.
19.
20. INSERT INTO Category
21. values (1,"Samsung Mobile Phone",400,"Samsung",5,3,4,"Mobile Phone comes with ch
    arger and headphone");
22.
23. INSERT INTO Category
```

```
values (2,"LG TV",2500,"LG",20,10,5,"Smart TV comes with remote and two AAA batteries");


INSERT INTO Mobiles
values (1,5,"Note 4");

INSERT INTO TV
values (2,43,"Smart ooo","LED");


INSERT INTO item
values (1,'2019-1-10 01:56:11',"Blue","Pending",'A','33',1,NULL,1);

INSERT INTO item
values (2,'2019-5-8 12:24:01',"Black","Orderd",'S','21',1,2,2);
```

UPDATING TABLES' TUPLES

```
UPDATE Entity set Country = "USA" WHERE ID = 1;
UPDATE Supplier set contract_start = '2019-5-7' WHERE ID = 1;
UPDATE Distributor set Tax_no = "48928347" WHERE ID = 2;
UPDATE Category set Company  = "Huawei" WHERE ID = 1;
UPDATE Mobile set Model_no = "P7" WHERE ID = 1;
```

DELETING TABLES' TUPLES

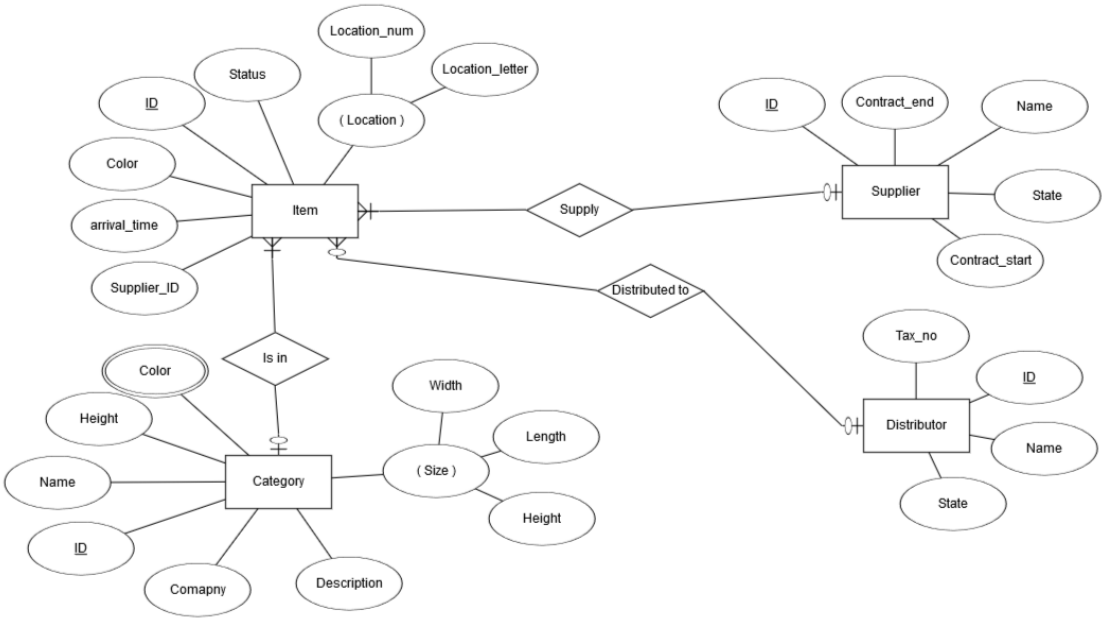
```
DELETE FROM Entity WHERE ID = 1;
DELETE FROM Supplier WHERE contract_start = '2019-5-7';
DELETE FROM Entity WHERE Tax_no = "48928347";
DELETE FROM Entity WHERE Company = "Huawei";
DELETE FROM Entity WHERE Model_no = "P7";
```

RETRIEVING FROM TABLES

```
SELECT * FROM Category WHERE ID = 2;

SELECT email, Tax_no -- select email and tax_no of all distributors from spain
FROM Entity, Distributor
WHERE Entity.ID = Distributor.ID
    AND Country = "spain";

SELECT contract_start, Name -- select contract date and name all distributors
FROM Entity, Distributor
WHERE Entity.ID = Distributor.ID;

SELECT Name, COUNT(*) -- number of huawei phones in the warehouse
FROM Mobile, Category
WHERE Mobile.ID = Category.ID
    AND Category.Company = "Huawei";

SELECT Company, COUNT(*) --
group phones by company and select all companies with more than 5 phones
FROM Mobile, Category
WHERE Mobile.ID = Category.ID
```

```
20. GROUP BY Category.Company
21. HAVING COUNT(*) > 5;
```

ER ERD lab Sample



Relational Model ERD lab Sample