

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL, COMPUTER AND COMMUNICATION

ENGINEERING

Database Design Project

Spring 2023

**STUDENT NO: 202110272**

**202110931**

**STUDENT NAME: Hassan Khadra**

**Youssef Eid**

1. A company has one or more locations referred to as "Plant" (Plant-ID). Several plants are uniquely assigned to the same company code (CC-ID).

* We created 2 entities: “Plant” and “Company Code”.
* “Plant” has the primary key [Plant-ID] and “Company Code” has the primary key [CC-ID].   
  The relation name between them is AssignedTo and it’s a 1:N relation.

1. Plants can have several storage locations (S-loc-#) in which the manufactured and purchased items are stocked, a storage location, however, can only be assigned to one plant.

* We created a new entity: “Storage Location”.
* “Storage Location” has the primary key [S-loc-#] and an attribute “Item”
* The relation name between “Storage Location” and “Plant” is AllocatedTo and it’s a 1:N relation.

1. A purchase organization (PO-ID) can be assigned to several plants and a plant to several purchase organizations. A company code is related to one or more purchase organizations, the latter is uniquely related to a company code.

* We created a new entity: “Purchase Organization”.
* “Purchase Organization” has the primary key [PO-ID].
* The relation name between “Purchase Organization” and “Plant” is DeterminedTo and it’s a N:M relation.
* The relation name between “Purchase Organization” and “Company Code” is RelatedTo and it’s a 1:N relation.

1. A distribution channel (DC-#) can be assigned to several divisions (Div-ID), whereas a division is uniquely assigned to a distribution channel.

* We created 2 entities: “Distribution Channel” and “Division”.
* “Distribution Channel” has the primary key [DC-#] and “Division” has the primary key [Div-ID].
* The relation name between them is DesignatedTo and it’s a 1:N relation.

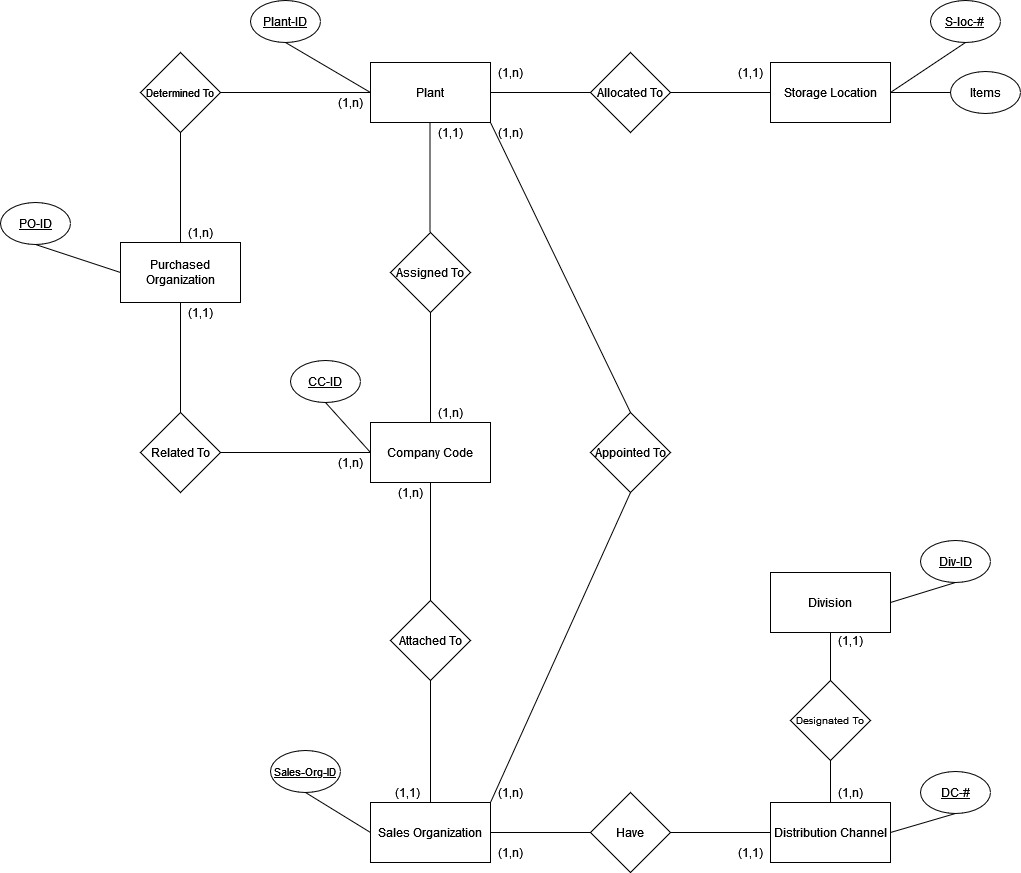
1. A sales organization (Sales-Org-ID) can have several distribution channels, whereas a distribution channel is uniquely assigned to a sales organization.

* We created a new entity: “Sales Organization”.
* “Sales Organization” has the primary key [Sales-Org-ID].
* The relation name between “Sales Organization” and “Distribution Channel” is have and it’s a 1:N relation.

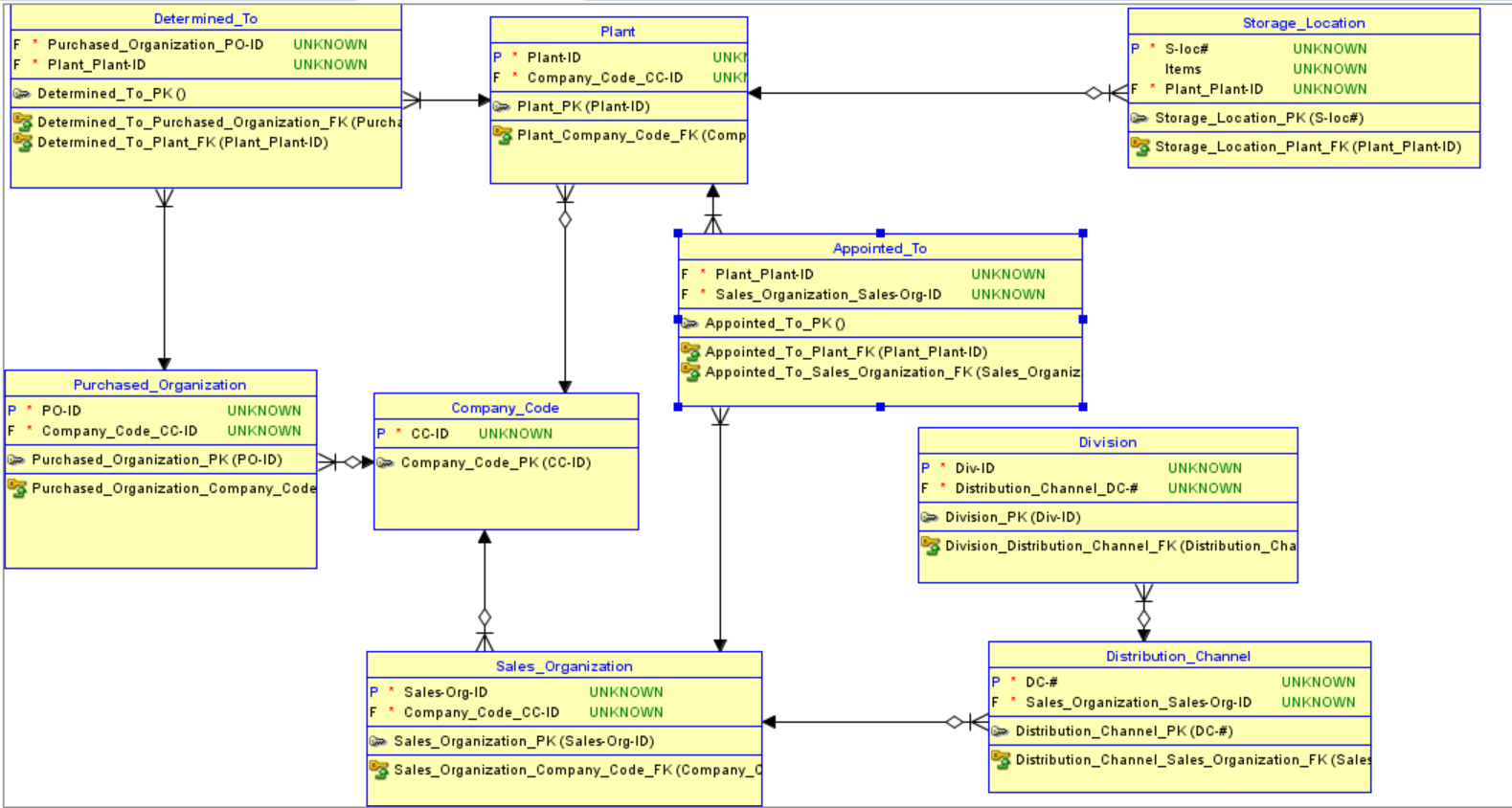
1. A company code can have several sales organizations. Sales organizations are assigned to one company code only and several plants. Plants can also be assigned to several sales organizations.

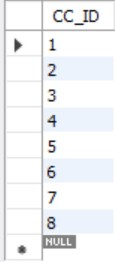
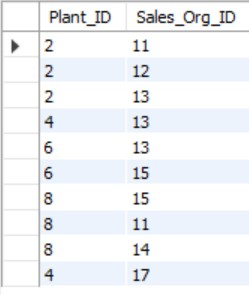
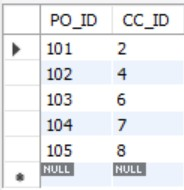
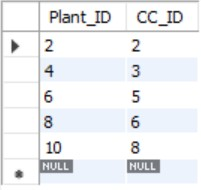
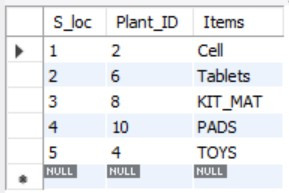
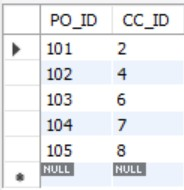
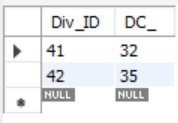
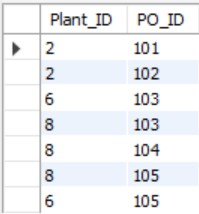
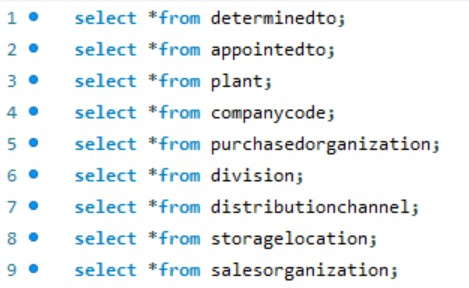
* The relation name between “Sales Organization” and “Company Code” is AttachedTo and it’s a 1:N relation.
* The relation name between “Sales Organization” and “Plant” is AppointedTo and it’s a N:M relation.

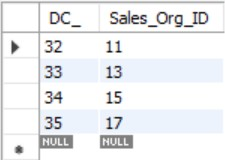
Conceptual Model:

****

Relational Model:



Physical Model:



**MySQL Queries:**

Creating the tables:

* Create table CompanyCode(CC\_ID int not null, primary key(CC\_ID));
* create table Plant(Plant\_ID int not null, CC\_ID int not null, primary key(Plant\_ID), foreign key (CC\_ID) references CompanyCode(CC\_ID));
* create table StorageLocation(S\_loc int not null, Plant\_ID int not null, Items varchar(50) null, primary key(S\_loc), foreign key (Plant\_ID) references Plant(Plant\_ID));
* create table PurchasedOrganization(PO\_ID int not null, CC\_ID int not null, primary key(PO\_ID), foreign key (CC\_ID) references CompanyCode(CC\_ID));
* create table SalesOrganization(Sales\_Org\_ID int not null, CC\_ID int not null, primary key(Sales\_Org\_ID), foreign key (CC\_ID) references CompanyCode(CC\_ID));
* create table DistributionChannel(DC\_ int not null, Sales\_Org\_ID int not null, primary key(DC\_), foreign key (Sales\_Org\_ID) references salesorganization(Sales\_Org\_ID));
* create table Division(Div\_ID int not null, DC\_ int not null, primary key(Div\_ID), foreign key (DC\_) references distributionchannel(DC\_));
* create table AppointedTo(Plant\_ID int not null, Sales\_Org\_ID int not null, foreign key(Plant\_ID) references plant(Plant\_ID), foreign key (Sales\_Org\_ID) references salesorganization(Sales\_Org\_ID));
* create table DeterminedTo(Plant\_ID int not null, PO\_ID int not null, foreign key(Plant\_ID) references plant(Plant\_ID), foreign key (PO\_ID) references purchasedorganization(PO\_ID));

Inserting into the tables (only showing 1 example for each table):

* insert into companycode(CC\_ID)values(1);
* insert into plant(Plant\_ID, CC\_ID)values(2, 2);
* insert into storagelocation(S\_loc, Plant\_ID, Items)values(1, 2,'Cell');
* insert into salesorganization(Sales\_Org\_ID, CC\_ID)values(11, 1);
* insert into distributionchannel(DC\_, Sales\_Org\_ID)values(32, 11);
* insert into division(Div\_ID, DC\_)values(41, 32);
* insert into appointedto(Plant\_ID,Sales\_Org\_ID)values(2, 11);
* insert into purchasedorganization(PO\_ID,CC\_ID)values(101,2);
* insert into determinedto(Plant\_ID, PO\_ID)values(2,101);

**Few MySql exercises:**

**1-a. Give the number of data in table Company Code having more than 3 records in table Sales Organization:**

* select count(\*) as Counter from companycode where (select salesorganization.CC\_ID from salesorganization where companycode.CC\_ID = salesorganization.CC\_ID group by salesorganization.CC\_ID having count(salesorganization.CC\_ID) >3);

**2-a. Show the data that occure in table Sales Organization having the maximum number of data in table Distribution Channel and order them by their count.**

* select salesorganization.Sales\_Org\_ID, salesorganization.CC\_ID from salesorganization, distributionchannel where distributionchannel.Sales\_Org\_ID = salesorganization.Sales\_Org\_ID group by salesorganization.Sales\_Org\_ID, salesorganization.CC\_ID order by count(\*);

The method we did is by selecting all data in table **Sales Organization** and order them by the number of times each data figured in table **Distribution Channel**.