



**Ain Shams University
Faculty of Engineering
Computers and Systems Engineering
Department**

CSE421: Database Systems

Trading Company

- An SQL Database Report

INTRODUCED BY

1601626

Neven Nabil Shokri

Contents

Database Design	4
Introduction:	4
Design Requirements.....	5
Entities and their attributes	5
Constraints	6
Assumptions	7
Diagrams	8
Detailed EER Diagram	9
Referential Diagram.....	10
SQL DDL.....	11
Declaring Relations.....	12
Create branch table.....	12
Create employee table	13
Create customer table	14
Create supplier table.....	15
Create supplies table	16
Create bill_in table.....	17
Create bill_body_in table	18
Create bill_out table.....	19
Create bill_body_out table	20
Create login table	21
Create role table	22
Create permissions table.....	23
Create emp_has table	24
Create product table	25
Create stock table	26
Create category table	27
Create sub_category table.....	28
Foreign Keys Constraints.....	29

Data insertion.....	31
Queries	34
Query 1: Cairo Employees who were born in the 90's	35
Statement:	35
Output :	35
Query 2: product sales in 2022	36
Statement :.....	36
Output :	36
Query 3: find suppliers supplies lives in Giza	37
Statement :.....	37
Output :	37
Query 4: all invoices(bills in) exceeds 1000L.E in 2022	38
Statement :.....	38
Output :	38
Query 5: all customers class A	39
Statement	39
Output :	39

Introduction:

Design

Trading companies are businesses working with different kinds of products which are sold for consumer, business, or government purposes. Trading companies buy a specialized range of products, maintain a stock or a shop, and deliver products to customers.

Importers or wholesalers maintain a stock and deliver products to shops or large end customers. They work in a large geographical area, while their customers, the shops, work in smaller areas and often in just a small neighborhood.

Design Requirements

I designed a database for a Trading Company System describes:

- the movement of the products into the stock from suppliers and from the stock to the customers
- Each movement considered as a bill with all required information
- either bill in from supplier (invoice) or bill out to customer (purchase)
- The system also manages the products and their categories
- manages information of employees' roles and permissions
- manages customers and suppliers' information and orders from suppliers to customers.
- some employees have permission to add, edit, delete data.
- the system also has stock where price and remaining quantity of each product are stored.

Entities and their attributes

- **company** (branch_id, name, street, postalcode, city).
- **employee** (person_id, FName, Minit, LName, phone, street, postalcode, city, gender, email, day, month, year, com_id, salary).
- **supplier** (person_id, FName, Minit, LName, phone, street, postalcode, city, gender, email, day, month, year, title).
- **customer** (person_id, FName, Minit, LName, phone, street, postalcode, city, gender, email, day, month, year, potential).
- **bill_in** (bill_id, bill_date, emp_id, bill_sup_id).
- **Bill_body_in** (bill_body_in_id, bill_in_id, prod_id, quantity, price, discount).
- **bill_out** (bill_id, bill_date, emp_id, bill_cus_id).
- **Bill_body_out** (bill_body_out_id, bill_out_id, prod_id, quantity, price, discount).
- **login** (id, emp_id, username, password).
- **role** (role_id, role_name, role_describtion).
- **permissions** (per_role_id, per_name).
- **emp_has** (emp_id, role_id, per_role_id).
- **product** (prod_id, prod_name, code, emp_id, ctg_id, subctg_id).
- **category** (ctg_id, ctg_name).
- **sub_category** (subctg_id, subctg_name).
- **supplies** (person_id, prod_id).
- **stock** (stock_id, prod_id, quantity, price).

Constraints

- Each person may be classified into employee or customer or supplier.
- Each branch must have many employees
- Each employee must belong to only one branch.
- Each employee may have one login
- Each login must belong to one employee
- Each employee must have one or more role
- Each role must go to one or more employee
- Each employee must have one or more permission
- Each permission must go to one or more employee
- Each supplier must supply one or more product.
- Each product must be supplied from one or more suppliers.
- Each employee may generate one or more bill.
- Each bill must be generated only by one employee.
- Each supplier may have many bill_in .
- Each customer may have many bill_out .
- Each bill out must be generated only by one customer.
- Each bill in must be generated only by one supplier.
- Each bill in must contain one or more bill_body_in
- Each bill out must contain one or more bill_body_out
- Each bill_body_in must belong to only one bill in .
- Each bill_body_out must belong to only one bill out .
- Each bill_body_in may update only one stock.
- Each bill_body_out may update only one stock.
- Each product must have one category
- Each product must have one sub category
- Each category may have many products

- Each sub category may have many products

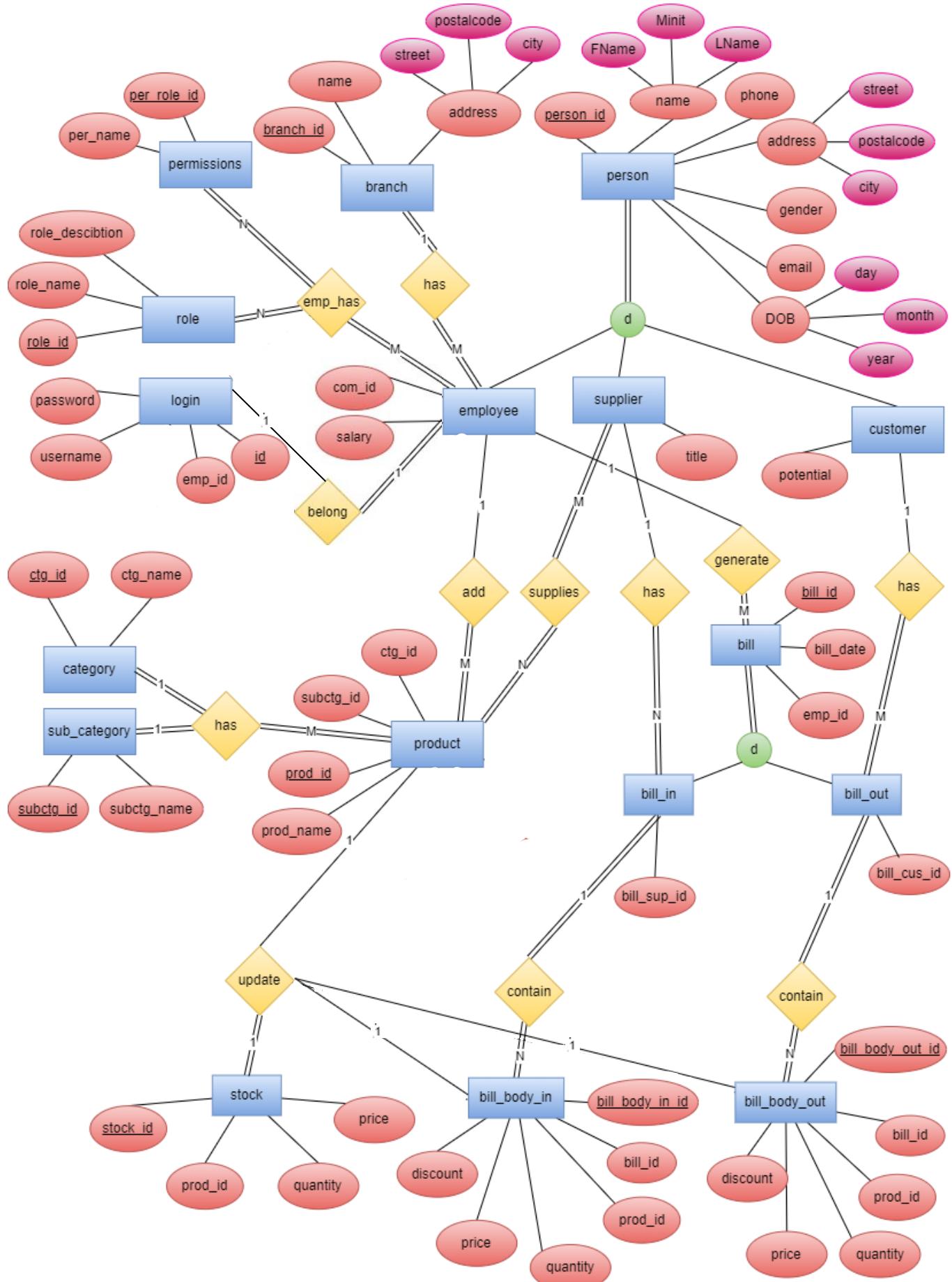
Assumptions

- Customers can't login nor suppliers only the employee can.
- The system doesn't support late payments, all paid cash and at the same date of the bill.
- All price in L.E

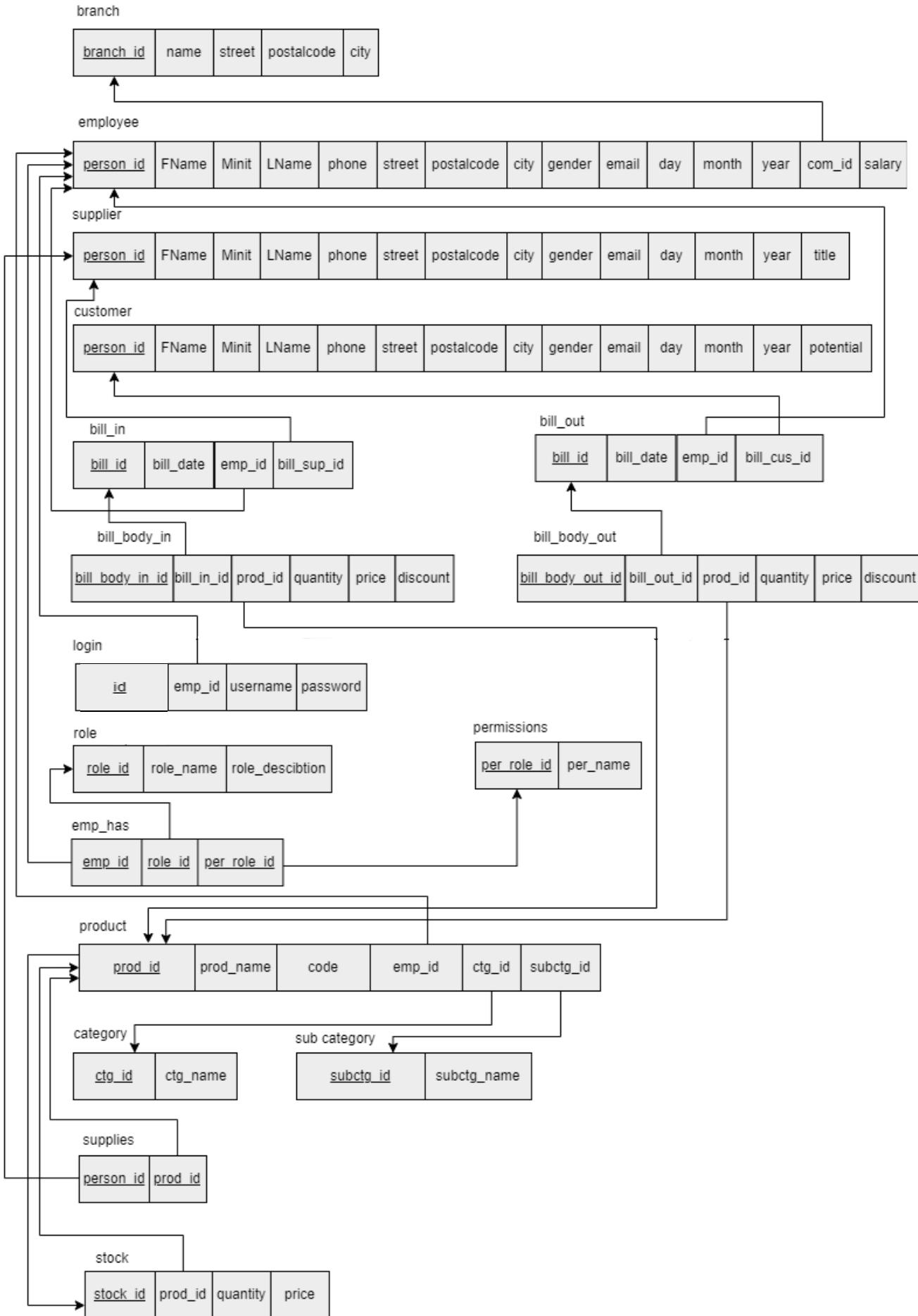
Database Diagrams

- Detailed EER Diagram
- Referential Diagram

Detailed EER Diagram



Referential Diagram



Database

SQL DDL

- Declaring Relations
- Foreign Keys Constraints

Declaring Relations

Create branch table

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for SQL, Text, Execute, Stop, Refresh, and others.
- Navigator:** Shows the schema `trading_company` with tables `bill_body_in`, `bill_body_out`, `bill_in`, `bill_out`, and `branch`. The `branch` table is selected.
- Query Editor:** Contains two CREATE TABLE statements:

```
1 • CREATE TABLE branch(
2     branch_id int NOT NULL AUTO_INCREMENT,
3     name VARCHAR(30) NOT NULL,
4     street VARCHAR(20),
5     postalcode VARCHAR(20),
6     city VARCHAR(20),
7     PRIMARY KEY(branch_id)
8 );
9
10 • CREATE TABLE employee(
11     person_id int NOT NULL AUTO_INCREMENT,
12     FName VARCHAR(20) NOT NULL,
13     Minit CHAR,
14     LName VARCHAR(20) NOT NULL,
15     phone VARCHAR(15) NOT NULL,
16     street VARCHAR(20),
17     postalcode VARCHAR(20),
18     city VARCHAR(20),
19     gender CHAR(1),
20     email VARCHAR(30),
21     day VARCHAR(10),
22     month VARCHAR(10),
```
- Output Panel:** Shows the execution log with three successful queries (IDs 50, 51, 52) and their execution times.
- Object Info:** Tab selected, showing information about the `branch` table.

Table: branch

Columns:

Column	Type
<code>branch_id</code>	int AI PK
<code>name</code>	varchar(30)
<code>street</code>	varchar(20)
<code>postalcode</code>	varchar(20)
<code>city</code>	varchar(20)

Action Output

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Create employee table

MySQL Workbench

localhost × unconnected × unconnected × unconnected ×

File Edit View Query Database Server Tools Scripting Help

Navigator: creat tables Forein_Keys insertion SQL File 21* ×

SCHEMAS

Filter objects

- Minit
- LName
- phone
- street
- postalcode
- city
- gender
- email
- day
- month
- year
- com_id
- salary
- login_id

Indexes

Foreign Keys

Administration Schemas

Information

Table: employee

Columns:

Column Name	Type
person_id	char(10) PK
FName	varchar(10)
Minit	char(1)
LName	varchar(20)
phone	varchar(10)
street	varchar(20)
postalcode	varchar(20)
city	varchar(20)
gender	char(1)
email	varchar(30)
day	varchar(10)
month	varchar(10)
year	varchar(10)
com_id	char(10)
salary	decimal(6,1)
login_id	char(10)

1 • CREATE TABLE employee(
2 person_id CHAR(10) NOT NULL,
3 FName VARCHAR(10) NOT NULL,
4 Minit CHAR,
5 LName VARCHAR(20) NOT NULL,
6 phone VARCHAR(10) NOT NULL,
7 street VARCHAR(20),
8 postalcode VARCHAR(20),
9 city VARCHAR(20),
10 gender CHAR(1),
11 email VARCHAR(30),
12 day VARCHAR(10),
13 month VARCHAR(10),
14 year VARCHAR(10),
15 com_id CHAR(10),
16 salary decimal(6,1),
17 login_id CHAR(10),
18 PRIMARY KEY(person_id)
19);

Object Info Session

Query Completed

Create customer table

MySQL Workbench

unconnected × localhost ×

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 9* SQL File 3* SQL File 6* SQL File 7* SQL File 8* SQLAdditions

SCHEMAS

CREATE TABLE customer(person_id CHAR(10) NOT NULL, FName VARCHAR(10) NOT NULL, Minit CHAR, LName VARCHAR(20) NOT NULL, phone VARCHAR(10) NOT NULL, street VARCHAR(20), postalcode VARCHAR(20), city VARCHAR(20), gender CHAR(1), email VARCHAR(30), day VARCHAR(10), month VARCHAR(10), year VARCHAR(10), potential VARCHAR(10), PRIMARY KEY(person_id)) ;

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Table: customer

Columns:

person_id	char(10) PK
FName	varchar(10)
Minit	char(1)
LName	varchar(20)
phone	varchar(10)
street	varchar(20)
postalcode	varchar(20)
city	varchar(20)
gender	char(1)
email	varchar(30)
day	varchar(10)
month	varchar(10)
year	varchar(10)
potential	varchar(10)

Object Info Session

Query Completed

Action Output

#	Time	Action	Message	Duration / Fetch
9	14:06:58	CREATE TABLE branch(branch_id CHAR(10) NOT NULL, FName VARCHAR(10) NOT NULL, Minit CHAR, LName VARCHAR(20) NOT NULL, phone VARCHAR(10) NOT NULL, street VARCHAR(20), postalcode VARCHAR(20), city VARCHAR(20), gender CHAR(1), email VARCHAR(30), day VARCHAR(10), month VARCHAR(10), year VARCHAR(10), potential VARCHAR(10), PRIMARY KEY(branch_id)) ;	0 row(s) affected	0.015 sec
10	14:07:51	CREATE TABLE employee(person_id CHAR(10) NOT NULL, FName VARCHAR(10) NOT NULL, Minit CHAR, LName VARCHAR(20) NOT NULL, phone VARCHAR(10) NOT NULL, street VARCHAR(20), postalcode VARCHAR(20), city VARCHAR(20), gender CHAR(1), email VARCHAR(30), day VARCHAR(10), month VARCHAR(10), year VARCHAR(10), potential VARCHAR(10), PRIMARY KEY(person_id)) ;	0 row(s) affected, 2 warning(s): 1681 Integer division by zero	0.016 sec
11	14:08:45	CREATE TABLE supplier(person_id CHAR(10) NOT NULL, FName VARCHAR(10) NOT NULL, Minit CHAR, LName VARCHAR(20) NOT NULL, phone VARCHAR(10) NOT NULL, street VARCHAR(20), postalcode VARCHAR(20), city VARCHAR(20), gender CHAR(1), email VARCHAR(30), day VARCHAR(10), month VARCHAR(10), year VARCHAR(10), potential VARCHAR(10), PRIMARY KEY(person_id)) ;	0 row(s) affected	0.031 sec
12	14:09:47	CREATE TABLE customer(person_id CHAR(10) NOT NULL, FName VARCHAR(10) NOT NULL, Minit CHAR, LName VARCHAR(20) NOT NULL, phone VARCHAR(10) NOT NULL, street VARCHAR(20), postalcode VARCHAR(20), city VARCHAR(20), gender CHAR(1), email VARCHAR(30), day VARCHAR(10), month VARCHAR(10), year VARCHAR(10), potential VARCHAR(10), PRIMARY KEY(person_id)) ;	0 row(s) affected	0.016 sec

Create supplier table

MySQL Workbench

localhost

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Supplier Columns

person_id FName Minit LName phone street postalcode city gender email day month year title

Table: supplier

Columns:

Column Name	Type	Properties
person_id	int AI PK	
FName	varchar(10)	
Minit	char(1)	
LName	varchar(20)	
phone	varchar(15)	
street	varchar(20)	
postalcode	varchar(20)	
city	varchar(20)	
gender	char(1)	
email	varchar(30)	
day	varchar(10)	
month	varchar(10)	
year	varchar(10)	
title	varchar(10)	

create tables Forien_Keys insertions Queries

46
47 • CREATE TABLE supplier(
48 person_id int NOT NULL AUTO_INCREMENT,
49 FName VARCHAR(10) NOT NULL,
50 Minit CHAR,
51 LName VARCHAR(20) NOT NULL,
52 phone VARCHAR(15) NOT NULL,
53 street VARCHAR(20),
54 postalcode VARCHAR(20),
55 city VARCHAR(20),
56 gender CHAR(1),
57 email VARCHAR(30),
58 day VARCHAR(10),
59 month VARCHAR(10),
60 year VARCHAR(10),
61 title VARCHAR(10),
62 PRIMARY KEY(person_id)
63);
64
65 • CREATE TABLE supplies(
66 person_id int NOT NULL,
67 prod_id int NOT NULL

Output Action Output

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session 6:50 PM

Create supplies table

MySQL Workbench

localhost

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

permissions
product
role
stock
sub_category
supplier
supplies
Columns
Indexes
ForeignKeys
Triggers
Views
Stored Procedures
Functions

creat tables insertions Forien_Keys Queries

64
65 • CREATE TABLE supplies(
66 person_id int NOT NULL,
67 prod_id int NOT NULL
68);
69
70 • CREATE TABLE bill_in(
71 bill_id int NOT NULL AUTO_INCREMENT,
72 bill_date date NOT NULL,
73 emp_id int NOT NULL,
74 bill_sup_id int NOT NULL,
75 PRIMARY KEY(bill_id)
76);
77
78 • CREATE TABLE bill_body_in(
79 bill_body_in_id int NOT NULL AUTO_INCREMENT,
80 bill_in_id int NOT NULL,
81 prod_id int NOT NULL,
82 quantity int(30) NOT NULL,
83 price decimal(6,1),
84 discount decimal(6,1) default 0.0,
85 PRIMARY KEY(bill_body_in_id)

Table: supplies

Columns:
person_id int
prod_id int

Related Tables:
product
Target (prod_id → prod_id)
On Update CASCADE
On Delete CASCADE
employee
Target (person_id → person_id)
On Update CASCADE
On Delete CASCADE
Object Info Session

Output

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Create bill_in table

MySQL Workbench

localhost ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

trading_company

Tables

- bill_body_in
- bill_body_out
- bill_in

Columns

- bill_id
- bill_date
- emp_id
- bill_sup_id

Indexes

Foreign Keys

Triggers

bill_out

branch

category

customer

Administration Schemas

Information

Table: bill_in

Columns:

bill_id	int AI PK
bill_date	date
emp_id	int
bill_sup_id	int

67 prod_id int NOT NULL
68) ;
69
70 • CREATE TABLE bill_in(
71 bill_id int NOT NULL AUTO_INCREMENT,
72 bill_date date NOT NULL,
73 emp_id int NOT NULL,
74 bill_sup_id int NOT NULL,
75 PRIMARY KEY(bill_id)
76) ;
77
78 • CREATE TABLE bill_body_in(
79 bill_body_in_id int NOT NULL AUTO_INCREMENT,
80 bill_in_id int NOT NULL,
81 prod_id int NOT NULL,
82 quantity int(30) NOT NULL,
83 price decimal(6,1),
84 discount decimal(6,1) default 0.0,
85 PRIMARY KEY(bill_body_in_id)
86) ;
87
88 • CREATE TABLE bill_out(
89
Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id,prod_id AS Product_id,bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

6:52 PM

Create bill_body_in table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Refresh, Undo, Redo, Save, Print, and others.
- Navigator:** Shows the schema **trading_company** with tables **bill_body_in**, **bill_body_out**, and **bill_in**.
- Current Editor:** The **create tables** tab is active, displaying the SQL code for creating three tables:

```
76  );
77
78 • CREATE TABLE bill_body_in(
79     bill_body_in_id int NOT NULL AUTO_INCREMENT,
80     bill_in_id int NOT NULL,
81     prod_id int NOT NULL,
82     quantity int(30) NOT NULL,
83     price decimal(6,1),
84     discount decimal(6,1) default 0.0,
85     PRIMARY KEY(bill_body_in_id)
86 );
87
88 • CREATE TABLE bill_out(
89     bill_id int NOT NULL AUTO_INCREMENT,
90     bill_date date NOT NULL,
91     emp_id int NOT NULL,
92     bill_cus_id int NOT NULL,
93     PRIMARY KEY(bill_id)
94 );
95
96 • CREATE TABLE bill_body_out(
97     bill_body_out_id int NOT NULL AUTO_INCREMENT,
```
- Output Tab:** Shows the **Action Output** table with three rows of log entries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec
- Bottom Status:** Shows the session status and the time 6:52 PM.

Create bill_out table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **SCHEMAS** section with the **trading_company** schema selected. Inside the schema, the **Tables** section lists **bill_body_in**, **bill_body_out**, **bill_in**, and **bill_out**. The **bill_out** table is currently selected.
- SQL Editor:** Displays the SQL code for creating the **bill_out** table. The code includes columns: **bill_id** (int NOT NULL AUTO_INCREMENT), **bill_date** (date NOT NULL), **emp_id** (int NOT NULL), and **bill_cus_id** (int NOT NULL). It also includes a primary key constraint on **bill_id**. The code is numbered from 85 to 106.
- Output:** Shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec
- Object Info:** Tab selected at the bottom left.
- Session:** Tab selected at the bottom right.

Create bill_body_out table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **SCHEMAS** section with **trading_company** selected. Under **Tables**, **bill_body_out** is selected, revealing its **Columns**: **bill_body_out_id**, **bill_out_id**, **prod_id**, **quantity**, **price**, and **discount**.
- Query Editor:** Displays the SQL code for creating the **bill_body_out** table and other tables like **login** and **role**. The code includes primary key definitions and data types.
- Output:** Shows the results of recent queries, including three SELECT statements that returned 1, 5, and 2 rows respectively, all completed in 0.000 sec.

Create login table

MySQL Workbench

localhost

File Edit View Query Database Server Tools Scripting Help

Navigator creat tables* insertions Forein_Keys Queries

SCHEMAS

Filter objects

106
107 • CREATE TABLE login(
108 id int NOT NULL AUTO_INCREMENT,
109 emp_id int NOT NULL,
110 username VARCHAR(20) ,
111 password VARCHAR(20) ,
112 PRIMARY KEY(id)
113);
114
115 • CREATE TABLE role(
116 role_id int NOT NULL AUTO_INCREMENT,
117 role_name VARCHAR(20) ,
118 role_descibtion VARCHAR(20) ,
119 PRIMARY KEY(role_id)bill_body_outlogin
120);
121
122 • CREATE TABLE permissions(
123 per_role_id int NOT NULL AUTO_INCREMENT,
124 per_name CHAR(20) NOT NULL,
125 PRIMARY KEY(per_role_id)
126);
127

Table: login

Columns:

id	int AI PK
emp_id	int
username	varchar(20)
password	varchar(20)

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Create role table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Navigator, Schemas, Tables, Insertions, Foreign Keys, and Queries.
- Navigator:** Shows the database schema with the 'role' table selected. The 'role' table has columns: role_id (int AI PK), role_name (varchar(20)), and role_descibtion (varchar(20)).
- Central Area:** A code editor window titled 'creat tables*' containing the SQL script for creating three tables: 'role', 'permissions', and 'emp_has'. The 'role' table is defined with columns role_id (int NOT NULL AUTO_INCREMENT), role_name (VARCHAR(20)), and role_descibtion (VARCHAR(20)), with PRIMARY KEY(role_id). The 'permissions' table is defined with columns per_role_id (int NOT NULL AUTO_INCREMENT), per_name (CHAR(20) NOT NULL), and PRIMARY KEY(per_role_id). The 'emp_has' table is defined with columns emp_id (int NOT NULL), role_id (int NOT NULL), and per_role_id (int NOT NULL).
- Output:** An 'Action Output' table showing the results of recent queries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec
- Bottom Status:** Shows the session status and the time 6:54 PM.

Create permissions table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Navigator, Schemas, Tables, Insertions, Foreign Keys, Queries, and various search and filter options.
- Navigator Panel:** Shows the database schema with the "permissions" table selected. It displays columns: per_role_id (int AI PK) and per_name (char(20)).
- Central Editor Area:** Displays the SQL code for creating the "permissions" table and its associated tables ("branch", "category", "customer", "emp_has", "employee", "login").

```
118     role_descibtion VARCHAR(20) ,
119     PRIMARY KEY(role_id)
120   );
121
122 • CREATE TABLE permissions(
123   per_role_id int NOT NULL AUTO_INCREMENT,
124   per_name CHAR(20) NOT NULL,
125   PRIMARY KEY(per_role_id)
126 );
127
128 • CREATE TABLE emp_has(
129   emp_id int NOT NULL,
130   role_id int NOT NULL,
131   per_role_id int NOT NULL
132 );
133
134 • CREATE TABLE product(
135   prod_id int NOT NULL AUTO_INCREMENT,
136   prod_name VARCHAR(20),
137   code VARCHAR(20) unique,
138   ctg_id int NOT NULL,
139   subctg_id int NOT NULL,
```
- Output Panel:** Shows the "Action Output" log with three entries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id,prod_id AS Product_id,bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec
- Bottom Status Bar:** Shows the session status and time (6:55 PM).

Create emp_has table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Navigator, Schemas, Tables, Columns, Indexes, Foreign Keys, Triggers, Employee, Login, Permissions, Product, Role, Stock, and a search bar.
- Navigator Panel:** Shows the database schema with the **emp_has** table selected. The **Columns** section lists **emp_id**, **role_id**, and **per_role_id**.
- Query Editor:** Displays the SQL code for creating the **emp_has**, **product**, and **stock** tables. The **emp_has** table has columns **emp_id**, **role_id**, and **per_role_id**. The **product** table has columns **prod_id**, **prod_name**, **code**, **ctg_id**, and **subctg_id**. The **stock** table has columns **stock_id** and **prod_id**.
- Output Panel:** Shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Create product table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **product** table selected.
- Query Editor:** Displays the SQL code for creating the **product**, **stock**, and **category** tables.
- Output:** Shows the execution log with three SELECT statements and their results.

```
130     role_id int NOT NULL,
131     per_role_id int NOT NULL
132   );
133
134 • ⓧ CREATE TABLE product(
135     prod_id int NOT NULL AUTO_INCREMENT,
136     prod_name VARCHAR(20),
137     code VARCHAR(20) unique,
138     ctg_id int NOT NULL,
139     subctg_id int NOT NULL,
140     PRIMARY KEY(prod_id)
141   );
142
143 • ⓧ CREATE TABLE stock(
144     stock_id int NOT NULL AUTO_INCREMENT,
145     prod_id int NOT NULL,
146     quantity int(255) NOT NULL,
147     price decimal(6,1),
148     PRIMARY KEY(stock_id)
149   );
150
151 • ⓧ CREATE TABLE category(
```

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Create stock table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **stock** table selected.
- Query Editor:** Displays the SQL code for creating three tables:

```
141    );
142
143 • CREATE TABLE stock(
144     stock_id int NOT NULL AUTO_INCREMENT,
145     prod_id int NOT NULL,
146     quantity int(255) NOT NULL,
147     price decimal(6,1),
148     PRIMARY KEY(stock_id)
149 );
150
151 • CREATE TABLE category(
152     ctg_id int NOT NULL AUTO_INCREMENT,
153     ctg_name VARCHAR(20),
154     PRIMARY KEY(ctg_id)
155 );
156
157 • CREATE TABLE sub_category(
158     subctg_id int NOT NULL AUTO_INCREMENT,
159     subctg_name VARCHAR(20),
160     PRIMARY KEY(subctg_id)
161 );
162
```
- Information:** Provides details about the **stock** table, including columns: **stock_id** (int AI PK), **prod_id** (int), **quantity** (int), and **price** (decimal(6,1)).
- Action Output:** Shows the results of three recent queries:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_Id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Create category table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Navigator, Schemas, Tables, Columns, Indexes, Foreign Keys, Triggers, Customer, Employee, Login, Permissions, Product, Role, Stock, Sub Categories, Insert, Select, Update, Delete, Refresh, and Save.
- Navigator Panel:** Shows the schema structure. Under the "category" schema, there are "Columns" (ctg_id, ctg_name), "Indexes", "Foreign Keys", and "Triggers". Other schemas listed include branch, customer, emp_has, employee, login, permissions, product, role, stock, and sub categories.
- Current Editor:** "creat tables" tab. The code being run is:

```
147     price decimal(6,1),
148     PRIMARY KEY(stock_id)
149   );
150
151 • CREATE TABLE category(
152     ctg_id int NOT NULL AUTO_INCREMENT,
153     ctg_name VARCHAR(20),
154     PRIMARY KEY(ctg_id)
155   );
156
157 • CREATE TABLE sub_category(
158     subctg_id int NOT NULL AUTO_INCREMENT,
159     subctg_name VARCHAR(20),
160     PRIMARY KEY(subctg_id)
161   );
162
163
164
165
166
167
168
```
- Output Panel:** Action Output table showing the results of the queries.

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec
- Bottom Status Bar:** Shows the date and time as 6:56 PM.

Create sub_category table

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with the **sub_category** table selected.
- Query Editor:** Displays the SQL code for creating the **sub_category** table:

```
154 PRIMARY KEY(ctg_id)
155 );
156
157 • CREATE TABLE sub_category(
158     subctg_id int NOT NULL AUTO_INCREMENT,
159     subctg_name VARCHAR(20),
160     PRIMARY KEY(subctg_id)
161 );
162
163
164
165
166
167
168
169
170
171
172
173
174
175
```
- Output:** Shows the execution history of three SELECT statements:

#	Time	Action	Message	Duration / Fetch
50	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	1 row(s) returned	0.015 sec / 0.000 sec
51	18:13:14	SELECT bill_id, prod_id AS Product_id, bill_d...	5 row(s) returned	0.016 sec / 0.000 sec
52	18:13:14	SELECT concat(FName, " ", LName) AS Sup...	2 row(s) returned	0.000 sec / 0.000 sec

Foreign Keys Constraints

- ```
ALTER TABLE employee
ADD CONSTRAINT emp Bran FOREIGN KEY (com_id) REFERENCES branch(branch_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE bill_in
ADD CONSTRAINT in_bill_emp FOREIGN KEY (emp_id) REFERENCES employee(person_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT in_bill_sup FOREIGN KEY (bill_sup_id) REFERENCES supplier(person_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE bill_body_in
ADD CONSTRAINT in_bodytohead FOREIGN KEY (bill_in_id) REFERENCES bill_in(bill_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT in_prod_bill FOREIGN KEY (prod_id) REFERENCES product(prod_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE bill_out
ADD CONSTRAINT out_bill_emp FOREIGN KEY (emp_id) REFERENCES employee(person_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT out_bill_cus FOREIGN KEY (bill_cus_id) REFERENCES customer(person_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE bill_body_out
ADD CONSTRAINT out_bodytohead FOREIGN KEY (bill_out_id) REFERENCES bill_out(bill_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT out_prod_bill FOREIGN KEY (prod_id) REFERENCES product(prod_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

- ```
ALTER TABLE login
ADD CONSTRAINT log_emp FOREIGN KEY (emp_id) REFERENCES employee(person_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE emp_has
ADD CONSTRAINT has_emp FOREIGN KEY (emp_id) REFERENCES employee(person_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT has_role FOREIGN KEY (role_id) REFERENCES role(role_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT has_per FOREIGN KEY (per_role_id) REFERENCES permissions(per_role_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE product
ADD CONSTRAINT prod_ctg FOREIGN KEY (ctg_id) REFERENCES category(ctg_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT prod_subCtg FOREIGN KEY (subctg_id) REFERENCES sub_category(subctg_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE supplies
ADD CONSTRAINT supp FOREIGN KEY (person_id) REFERENCES employee(person_id)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT sup_prod FOREIGN KEY (prod_id) REFERENCES product(prod_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```
- ```
ALTER TABLE stock
ADD CONSTRAINT stock_id FOREIGN KEY (prod_id) REFERENCES product(prod_id)
ON DELETE CASCADE ON UPDATE CASCADE;
```

Data insertion

- ```
INSERT INTO branch (name,street ,postalcode ,city)
VALUES ('Alex Branch', '638','El-Marg', 'October'),
('Giza Branch', '152','El-Saf', 'Giza'),
('Cairo Branch', '731', 'Ain Shams', 'Cairo');
```
- ```
INSERT INTO employee (FName, Minit,LName ,phone ,street ,postalcode ,city ,gender ,email ,day ,month ,year, com_id, salary )
VALUES ('Niveen','K','Adel', '01258965458','40','Sarayat','Cairo','F','neven40@gmail.com','20','3','1992','1','4000'),
('Islam','T','Magdi', '01525469523','36','Khalifa','Zagazig','M','islaam@gmail.com','11','5','1979','3','3500'),
('Alaa','S','Mohammed', '01025436598','12','El-Tahrir','Giza','F','Alaa399@gmail.com','14','7','1991','1','4500'),
('Ibrahim','K','Mohsen', '01232532697','20','Fire El-Nasser','Cairo','M','Ibrahim38@gmail.com','3','1','1996','2','6000'),
('Ali','A','Fikry', '01025125463','14','Ain Shams','Cairo','M','AliFikry200@gmail.com','12','2','1986','3','7500');
```
- ```
INSERT INTO login (emp_id,username ,password)
VALUES ('1', 'Niveen123','gkj256h21'),
('2', 'Islam345','ert25vh'),
('3', 'Alaa125','dffr56hbb21'),
('4', 'Ibrahim654','wer12f52'),
('5', 'Ali1267','drt12g5');
```
- ```
INSERT INTO supplier (FName, Minit,LName ,phone ,street ,postalcode ,city ,gender ,email ,day ,month ,year, title )
VALUES ('Soaad','K','Adel', '01258965458','40','Sarayat','Cairo','F','Soaad40@gmail.com','20','3','1992','chemicals'),
('Omar','T','Magdi', '01525469523','36','Khalifa','Zagazig','M','Omar@gmail.com','11','5','1979','fashion'),
('Mary','S','Mohammed', '01025436598','12','El-Tahrir','Giza','F','Mary399@gmail.com','14','7','1990','chemicals'),
('Fathy','K','Mohsen', '01232532697','20','Fire El-Nasser','Cairo','M','Fathy38@gmail.com','3','1','1996','furniture'),
('Ali','A','Fikry', '01025125463','14','Ain Shams','Cairo','M','AliFikry200@gmail.com','12','2','1986','fashion');
```
- ```
INSERT INTO customer (FName, Minit,LName ,phone ,street ,postalcode ,city ,gender ,email ,day ,month ,year, potential)
VALUES ('Hanaa','K','Adel', '01258965458','40','Sarayat','Cairo','F','Hanaa40@gmail.com','20','3','1992','A'),
('Magdy','T','Magdi', '01525469523','36','Khalifa','Zagazig','M','Magdy@gmail.com','11','5','1979','B'),
('Mariam','S','Mohammed', '01025436598','12','El-Tahrir','Giza','F','Mariam399@gmail.com','14','7','1990','B'),
('Mina','K','Mohsen', '01232532697','20','Fire El-Nasser','Cairo','M','Mina38@gmail.com','3','1','1996','C'),
('nabil','A','Fikry', '01025125463','14','Ain Shams','Cairo','M','nabil200@gmail.com','12','2','1986','A');
```
- ```
INSERT INTO role (role_name, role_descibtion)
VALUES ('manager', 'contrl,lead'),
('employee', 'make bills'),
('worker', 'transporting goods');
```
- ```
INSERT INTO permissions (per_name)
VALUES ('burchasing'),
('selling'),
('make reports'),
('add product'),
('add employee'),
('add supplier'),
('add customer'),
('update price');
```

- ```
INSERT INTO emp_has ( emp_id, role_id,per_role_id)
VALUES ('2', '1','4'),
('2', '1','5'),
('2', '1','6'),
('2', '1','7'),
('1', '2','8'),
('5', '2','1'),
('5', '2','2'),
('3', '2','2'),
('1', '2','2'),
('4', '2','2'),
('4', '2','3');
```
- ```
INSERT INTO category (ctg_name)
VALUES ('body care'),
('fashion');
```
- ```
INSERT INTO sub_category (subctg_name)
VALUES ('nails'),
('cosmetics'),
('makeup'),
('dress'),
('t-shirt'),
('jeans');
```
- ```
INSERT INTO product (prod_name,code , ctg_id,subctg_id)
VALUES ('nail remover', 'N001','1', '1'),
('blusher', 'g201','1', '3'),
('shower', 'r569','1', '2'),
('bodyLoshn', 'u589','1', '2'),
('maskara', 'g256', '1', '3'),
('highWaste', 'yt569','2', '6'),
('shortDress', 'f256','2', '4'),
('nightDress', 'q236', '2', '4');
```
- ```
INSERT INTO stock ( prod_id,quantity ,price)
VALUES ('1', '100','70'),
('2', '150','150'),
('3', '141','130'),
('4', '351','140'),
('5', '120','250'),
('6', '213','400'),
('7', '250','450'),
('8', '400','750');
```
- ```
INSERT INTO bill_in (bill_date , emp_id,bill_sup_id)
VALUES ('2021-09-05', '1','3'),
('2021-12-03', '1','4'),
('2021-06-03', '3','2'),
('2022-10-02', '4','5'),
('2022-2-13', '5','1');
```

- ```
INSERT INTO bill_body_in (bill_in_id,prod_id ,quantity ,price ,discount )
VALUES ('1', '3','26', '130','5'),
('1', '5','13', '120','5'),
('2', '4','5', '140','0'),
('3', '7','22', '450','10'),
('3', '8','14', '750','0'),
('4', '2','2', '150','8'),
('5', '1','13', '70','0'),
('5', '6','20', '400','6');
```
- ```
INSERT INTO bill_out (bill_date , emp_id,bill_cus_id)
VALUES ('2021-09-05', '1','3'),
('2021-12-03', '1','4'),
('2021-06-03', '3','2'),
('2022-10-02', '4','5'),
('2022-2-13', '5','1');
```
- ```
INSERT INTO bill_body_out (bill_out_id,prod_id ,quantity ,price ,discount )
VALUES ('1', '3','12', '130','5'),
('1', '5','13', '35','5'),
('2', '4','5', '33','0'),
('3', '7','22', '122','10'),
('3', '8','14', '16','0'),
('4', '2','2', '54','8'),
('5', '1','13', '33','0'),
('5', '6','20', '22','6');
```
- ```
INSERT INTO supplies (person_id, prod_id)
VALUES ('1', '5'),
('2', '8'),
('5', '7'),
('3', '4'),
('4', '3'),
('5', '6'),
('3', '1'),
('1', '2'),
('2', '6');
```

Queries

# Queries

# Query 1: Cairo Employees who were born in the 90's

List of all employees

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • SELECT * FROM new_schema.employee;
```

The result grid displays the following data:

|   | person_id | FName | Minit    | LName | phone       | street | postalcode     | city    | gender | email                 | day  | month | year | com_id | salary |
|---|-----------|-------|----------|-------|-------------|--------|----------------|---------|--------|-----------------------|------|-------|------|--------|--------|
| 1 | Niveen    | K     | Adel     |       | 01258965458 | 40     | Sarayat        | Cairo   | F      | neven40@gmail.com     | 20   | 3     | 1992 | 1      | 4000.0 |
| 2 | Islam     | T     | Magdi    |       | 01525469523 | 36     | Khalifa        | Zagazig | M      | islaam@gmail.com      | 11   | 5     | 1979 | 3      | 3500.0 |
| 3 | Alaa      | S     | Mohammed |       | 01025436598 | 12     | El-Tahrir      | Giza    | F      | Alaa399@gmail.com     | 14   | 7     | 1991 | 1      | 4500.0 |
| 4 | Ibrahim   | K     | Mohsen   |       | 01232532697 | 20     | Fire El-Nasser | Cairo   | M      | Ibrahim38@gmail.com   | 3    | 1     | 1996 | 2      | 6000.0 |
| 5 | Ali       | A     | Flkry    |       | 01025125463 | 14     | Ain Shams      | Cairo   | M      | AliFlkry200@gmail.com | 12   | 2     | 1986 | 3      | 7500.0 |
| * | HULL      | HULL  | HULL     | HULL  | HULL        | HULL   | HULL           | HULL    | HULL   | HULL                  | HULL | HULL  | HULL | HULL   | HULL   |

Statement:

```
-- Q1
/*Retrieve all employee who lives in cairo and were born in the 90's
*/
SELECT concat(FName, " ", LName) AS Employee_name, concat(day, "-", month, "-", year)
AS Birth_date, concat(street, ", ", postalcode, ", ", city) AS Address
FROM employee
WHERE city LIKE '%Cairo%' AND year LIKE '199%' -- in the 90's
ORDER BY Employee_name;
```

Output :

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
9
SELECT concat(FName, " ", LName) AS Employee_name, concat(day, "-", month, "-", year)
AS Birth_date, concat(street, ", ", postalcode, ", ", city) AS Address
FROM employee
WHERE city LIKE '%Cairo%' AND year LIKE '199%' -- in the 90's
ORDER BY Employee_name;
```

The result grid displays the following data:

|   | Employee_name  | Birth_date | Address                 |
|---|----------------|------------|-------------------------|
| ▶ | Ibrahim Mohsen | 3-1-1996   | 20,Fire El-Nasser,Cairo |
|   | Niveen Adel    | 20-3-1992  | 40,Sarayat,Cairo        |

## Query 2: product sales in 2022

List of all bills out and products

The image shows three separate MySQL query results side-by-side. The first query (SELECT \* FROM trading\_company.bill\_out) returns 5 rows of data with columns bill\_id, bill\_date, emp\_id, and bill\_cus\_id. The second query (SELECT \* FROM trading\_company.bill\_body\_out) returns 8 rows of data with columns bill\_body\_out\_id, bill\_out\_id, prod\_id, quantity, price, and discount. The third query (SELECT \* FROM trading\_company.product) returns 8 rows of data with columns prod\_id, prod\_name, code, ctg\_id, and subctg\_id.

| bill_id | bill_date  | emp_id | bill_cus_id |
|---------|------------|--------|-------------|
| 1       | 2021-09-05 | 1      | 3           |
| 2       | 2021-12-03 | 1      | 4           |
| 3       | 2021-06-03 | 3      | 2           |
| 4       | 2022-10-02 | 4      | 5           |
| 5       | 2022-02-13 | 5      | 1           |
| *       | HULL       | HULL   | HULL        |

| bill_body_out_id | bill_out_id | prod_id | quantity | price | discount |
|------------------|-------------|---------|----------|-------|----------|
| 1                | 1           | 3       | 12       | 130.0 | 5.0      |
| 2                | 1           | 5       | 13       | 35.0  | 5.0      |
| 3                | 2           | 4       | 5        | 33.0  | 0.0      |
| 4                | 3           | 7       | 22       | 122.0 | 10.0     |
| 5                | 3           | 8       | 14       | 16.0  | 0.0      |
| 6                | 4           | 2       | 2        | 54.0  | 8.0      |
| 7                | 5           | 1       | 13       | 33.0  | 0.0      |
| 8                | 5           | 6       | 20       | 22.0  | 6.0      |
| *                | HULL        | HULL    | HULL     | HULL  | HULL     |

| prod_id | prod_name    | code  | ctg_id | subctg_id |
|---------|--------------|-------|--------|-----------|
| 1       | nail remover | N001  | 1      | 1         |
| 2       | blusher      | g201  | 1      | 3         |
| 3       | shower       | r569  | 1      | 2         |
| 4       | bodyLoshn    | u589  | 1      | 2         |
| 5       | maskara      | g256  | 1      | 3         |
| 6       | highWaste    | yt569 | 2      | 6         |
| 7       | shortDress   | f256  | 2      | 4         |
| 8       | nightDress   | q236  | 2      | 4         |
| *       | HULL         | HULL  | HULL   | HULL      |

(join 3 tables with INNER JOIN method)

Retrieve bill number, product id, Product\_name, order date, quantity, and the total price that occurred after the start of 2022

Statement :

```
-- Q2
/*Retrieve bill number, product id, Product_name, order date, quantity, and the total
price that occurred after the start of 2022
*/
SELECT bill_id, product.prod_id AS Product_id, prod_name AS Product_name,bill_date AS Order_date, quantity,
 ((quantity * price)-discount) AS Total_price
FROM bill_out INNER JOIN bill_body_out
 ON bill_id = bill_out_id
 INNER JOIN product
 ON bill_body_out.prod_id = product.prod_id
WHERE bill_date >= '2022-01-01'
ORDER BY Total_price DESC;
```

Output :

The image shows a single MySQL query result grid with the following data:

| bill_id | Product_id | Product_name | Order_date | quantity | Total_price |
|---------|------------|--------------|------------|----------|-------------|
| 5       | 6          | highWaste    | 2022-02-13 | 20       | 434.0       |
| 5       | 1          | nail remover | 2022-02-13 | 13       | 429.0       |
| 4       | 2          | blusher      | 2022-10-02 | 2        | 100.0       |

## Query 3: find suppliers supplies lives in Giza

list of all suppliers

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • SELECT * FROM new_schema.supplier;
```

The result grid displays the following data:

|   | person_id | FName | Minit | LName    | phone       | street | postalcode     | city    | gender | email                 | day  | month | year | title     |
|---|-----------|-------|-------|----------|-------------|--------|----------------|---------|--------|-----------------------|------|-------|------|-----------|
| ▶ | 1         | Soaad | K     | Adel     | 01258965458 | 40     | Sarayat        | Cairo   | F      | Soaad40@gmail.com     | 20   | 3     | 1992 | chemicals |
|   | 2         | Omar  | T     | Magdi    | 01525469523 | 36     | Khalifa        | Zagazig | M      | Omar@gmail.com        | 11   | 5     | 1979 | fashion   |
|   | 3         | Mary  | S     | Mohammed | 01025436598 | 12     | El-Tahrir      | Giza    | F      | Mary399@gmail.com     | 14   | 7     | 1990 | chemicals |
|   | 4         | Fathy | K     | Mohsen   | 01232532697 | 20     | Fire El-Nasser | Cairo   | M      | Fathy38@gmail.com     | 3    | 1     | 1996 | furniture |
| * | 5         | Ali   | A     | Fikry    | 01025125463 | 14     | Ain Shams      | Cairo   | M      | Alifikry200@gmail.com | 12   | 2     | 1986 | fashion   |
| * | HULL      | HULL  | HULL  | HULL     | HULL        | HULL   | HULL           | HULL    | HULL   | HULL                  | HULL | HULL  | HULL | HULL      |

Statement :

```
-- Q3
/*Retrieve all suppliers lives in Giza
*/
SELECT concat(FName, " ", LName) AS Supplier_name, concat(street, ", ", postalcode, ", ", city) AS Address
FROM supplier
WHERE city LIKE '%Giza%'
ORDER BY Supplier_name;
```

Output :

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is the same as above, but the results are filtered by city.

The result grid displays the following data:

|   | Supplier_name | Address           |
|---|---------------|-------------------|
| ▶ | Mary Mohammed | 12,El-Tahrir,Giza |

# Query 4: all invoices(bills in) exceeds 1000L.E in 2022

List of all bills in and products

The screenshot shows three separate result grids side-by-side. The first grid (bill\_in) has columns: bill\_id, bill\_date, emp\_id, bill\_sup\_id. The second grid (bill\_body\_in) has columns: bill\_body\_in\_id, bill\_in\_id, prod\_id, quantity, price, discount. The third grid (product) has columns: prod\_id, prod\_name, code, ctg\_id, subctg\_id. The bill\_in grid contains 5 rows. The bill\_body\_in grid contains 8 rows. The product grid contains 8 rows.

| bill_id | bill_date  | emp_id | bill_sup_id |
|---------|------------|--------|-------------|
| 1       | 2021-09-05 | 1      | 3           |
| 2       | 2021-12-03 | 1      | 4           |
| 3       | 2021-06-03 | 3      | 2           |
| 4       | 2022-10-02 | 4      | 5           |
| 5       | 2022-02-13 | 5      | 1           |
| *       | HULL       | HULL   | HULL        |

| bill_body_in_id | bill_in_id | prod_id | quantity | price | discount |
|-----------------|------------|---------|----------|-------|----------|
| 1               | 1          | 3       | 26       | 130.0 | 5.0      |
| 2               | 1          | 5       | 13       | 120.0 | 5.0      |
| 3               | 2          | 4       | 5        | 140.0 | 0.0      |
| 4               | 3          | 7       | 22       | 450.0 | 10.0     |
| 5               | 3          | 8       | 14       | 750.0 | 0.0      |
| 6               | 4          | 2       | 2        | 150.0 | 8.0      |
| 7               | 5          | 1       | 13       | 70.0  | 0.0      |
| 8               | 5          | 6       | 20       | 400.0 | 6.0      |
| *               | HULL       | HULL    | HULL     | HULL  | HULL     |

| prod_id | prod_name    | code  | ctg_id | subctg_id |
|---------|--------------|-------|--------|-----------|
| 1       | nail remover | N001  | 1      | 1         |
| 2       | blusher      | g201  | 1      | 3         |
| 3       | shower       | r569  | 1      | 2         |
| 4       | bodyLoshn    | u589  | 1      | 2         |
| 5       | maskara      | g256  | 1      | 3         |
| 6       | highWaste    | y1569 | 2      | 6         |
| 7       | shortDress   | f256  | 2      | 4         |
| 8       | nightDress   | q236  | 2      | 4         |
| *       | HULL         | HULL  | HULL   | HULL      |

(join 3 tables with INNER JOIN method)

Retrieve bill number, product id, Product\_name, order date, quantity, and the total price of orders exceeds 1000L.E.

Statement :

```
-- Q4
/*Retrieve bill number, product id, order date, Product_name, quantity, and the total
price of orders exceeds 1000L.E.
*/
SELECT bill_id, product.prod_id AS Product_id, prod_name AS Product_name, bill_date AS Order_date, quantity,
((quantity * price)-discount) AS Total_price
FROM bill_out INNER JOIN bill_body_in
ON bill_id = bill_in_id
INNER JOIN product
ON bill_body_in.prod_id = product.prod_id
WHERE ((quantity * price)-discount) >= '1000'
ORDER BY Total_price DESC;
```

Output :

The screenshot shows a single result grid with the following columns: bill\_id, Product\_id, Product\_name, Order\_date, quantity, Total\_price. The grid contains 5 rows of data.

| bill_id | Product_id | Product_name | Order_date | quantity | Total_price |
|---------|------------|--------------|------------|----------|-------------|
| 3       | 8          | nightDress   | 2021-06-03 | 14       | 10500.0     |
| 3       | 7          | shortDress   | 2021-06-03 | 22       | 9890.0      |
| 5       | 6          | highWaste    | 2022-02-13 | 20       | 7994.0      |
| 1       | 3          | shower       | 2021-09-05 | 26       | 3375.0      |
| 1       | 5          | maskara      | 2021-09-05 | 13       | 1555.0      |

## Query 5: all customers class A

All customers

| person_id | FName  | Minit | LName    | phone       | street | postalcode     | city    | gender | email               | day  | month | year | potential |
|-----------|--------|-------|----------|-------------|--------|----------------|---------|--------|---------------------|------|-------|------|-----------|
| 1         | Hanaa  | K     | Adel     | 01258965458 | 40     | Sarayat        | Cairo   | F      | Hanaa40@gmail.com   | 20   | 3     | 1992 | A         |
| 2         | Magdy  | T     | Magdi    | 01525469523 | 36     | Khalifa        | Zagazig | M      | Magdy@gmail.com     | 11   | 5     | 1979 | B         |
| 3         | Mariam | S     | Mohammed | 01025436598 | 12     | El-Tahrir      | Giza    | F      | Mariam399@gmail.com | 14   | 7     | 1990 | B         |
| 4         | Mina   | K     | Mohsen   | 01232532697 | 20     | Fire El-Nasser | Cairo   | M      | Mina38@gmail.com    | 3    | 1     | 1996 | C         |
| 5         | nabil  | A     | Fikry    | 01025125463 | 14     | Ain Shams      | Cairo   | M      | nabil200@gmail.com  | 12   | 2     | 1986 | A         |
| *         | HULL   | HULL  | HULL     | HULL        | HULL   | HULL           | HULL    | HULL   | HULL                | HULL | HULL  | HULL | HULL      |

Statement

```
-- Q5
/*Retrieve all customers class A
*/
SELECT concat(FName, " ", LName) AS Supplier_name,
concat(street, ", ", postalcode, ", ", city) AS Address,
gender, phone, email, potential
FROM customer
WHERE potential LIKE 'A'
ORDER BY Supplier_name;
```

Output :

| Supplier_name | Address            | gender | phone       | email              | potential |
|---------------|--------------------|--------|-------------|--------------------|-----------|
| Hanaa Adel    | 40,Sarayat,Cairo   | F      | 01258965458 | Hanaa40@gmail.com  | A         |
| nabil Fikry   | 14,Ain Shams,Cairo | M      | 01025125463 | nabil200@gmail.com | A         |