



Department of Engineering & Mathematics

## **INTRODUCTION TO DATABASES AND BIG DATA**

### **ASSIGNMENT-2**

#### **MERILL HAND TOOL**

Tutors:

Dr Elizabeth Uruchurtu  
James Baldwin

Student:

Mahendra Bairi  
310131490

## Table of Contents

<b>1. Introduction:</b>	3
<b>2. Final E-R Diagram:</b>	3
<b>2.1 Assumptions:</b>	4
<b>3. Merill hand tool tables create statements</b>	5
<b>4. Data Dictionary Information</b>	11
<b>5. Demonstration of SQL Queries</b>	16
<b>6. Conclusion:</b>	31

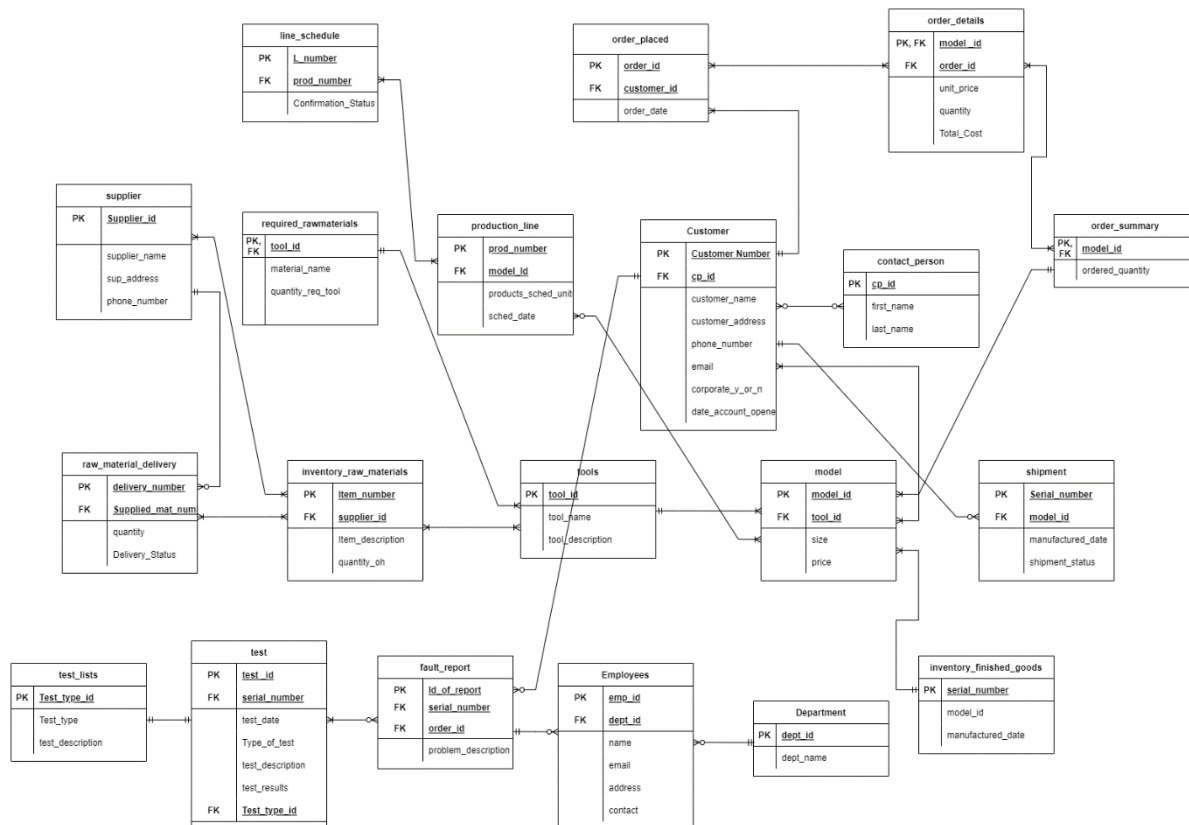
## 1. Introduction:

Merrill Hand Tools, located in Sheffield, is a hand tool manufacturer. Mike and Linda Merrill founded the business in 1980. They also brought a wealth of forging experience, not only from Sheffield but from all around the world.

MHT needed to expand its facilities at the turn of the century. Product Development, Manufacturing, and Marketing and Sales are MHT's three primary departments. As a result, MHT management has chosen this reengineering project. The new project is built on a client/server basis, with one or more databases connected to the internet.

Mainly this assignment is given to create a database as per company requirement with our assumptions by using Mysql workbench.

## 2. Final E-R Diagram:



The following changes I made form the feedback, I got in assignment1:

- Here I didn't take delivery company for to delivery tools to customers.
- And test and test\_lists I took separately.
- I used serial number according to database usage.
- Suplier and Raw\_material\_delivery tables are separate.

## **2.1 Assumptions:**

- Each employee is assigned to a single division.
- For few customers had a contact person from company.
- Order placed and order details I took as separate table to display data clear.
- Every regular supplier has supplier id.
- Test and test lists are separated and provided different id; test lists have a test type with test type id.
- Fault report was submitted by customers which is for delivered tool and that will be forwarded to employee table.
- A tool has a different model and when same models have a different number, so I considered tool id, model id, serial number separately.
- Some of the data I took from case study document and rest I filled fake data for below tables to generate a queries

### 3. Merrill hand tool tables create statements

#### Contact person:

```
CREATE TABLE `contact_person` (  
  `cp_id` int NOT NULL,  
  `first_name` varchar(45) NOT NULL,  
  `last_name` varchar(45) NOT NULL,  
  PRIMARY KEY (`cp_id`)  
);
```

#### Customer:

```
CREATE TABLE `customer` (  
  `Customer_number` varchar(10) NOT NULL,  
  `customer_name` varchar(45) NOT NULL,  
  `customer_address` varchar(80) NOT NULL,  
  `phone_number` varchar(15) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `corporate_y_or_n` varchar(5) NOT NULL,  
  `date_account_opened` date NOT NULL,  
  `cp_id` int DEFAULT NULL,  
  PRIMARY KEY (`Customer_number`),  
  KEY `cp_id_idx` (`cp_id`),  
  CONSTRAINT `cp_id` FOREIGN KEY (`cp_id`) REFERENCES `contact_person` (`cp_id`) ON UPDATE CASCADE  
);
```

#### Inventory\_raw\_materials:

```
CREATE TABLE `inventory_raw_materials` (  
  `Item_number` int NOT NULL,  
  `item_description` varchar(45) DEFAULT NULL,  
  `quantity_oh` int NOT NULL,  
  `supplier_id` int NOT NULL,  
  PRIMARY KEY (`Item_number`),  
  KEY `Sup_id_idx` (`supplier_id`),  
  CONSTRAINT `supply_id` FOREIGN KEY (`supplier_id`) REFERENCES `supplier` (`Supplier_id`) ON UPDATE CASCADE  
);
```

#### raw materials delivery:

```
CREATE TABLE `raw_mat_delivery` (  
  `delivery_number` int NOT NULL,  
  `Supplied_mat_number` int NOT NULL,  
  `quantity` int NOT NULL,
```

```

`Delivery_Status` varchar(10) NOT NULL,
PRIMARY KEY (`delivery_number`),
KEY `Item_number_idx` (`Supplied_mat_number`),
CONSTRAINT `Item_number` FOREIGN KEY (`Supplied_mat_number`) REFERENCES `inventory_raw_materials`
(`Item_number`) ON UPDATE CASCADE
);

```

### Supplier:

```

CREATE TABLE `supplier` (
  `Supplier_id` int NOT NULL,
  `supplier_name` varchar(45) NOT NULL,
  `sup_address` varchar(80) NOT NULL,
  `phone_number` varchar(11) NOT NULL,
  PRIMARY KEY (`Supplier_id`)
);

```

### Inventory finished goods:

```

CREATE TABLE `inventory_finished_goods` (
  `serial_number` int NOT NULL,
  `model_id` varchar(10) NOT NULL,
  `manufactured_date` date NOT NULL,
  PRIMARY KEY (`serial_number`)
);

```

### Shipment:

```

CREATE TABLE `shipment` (
  `Serial_number` int NOT NULL,
  `model_id` varchar(10) NOT NULL,
  `manufactured_date` date NOT NULL,
  `shipment_status` varchar(60) DEFAULT NULL,
  PRIMARY KEY (`Serial_number`),
  KEY `m_id_idx` (`model_id`),
  CONSTRAINT `m_id` FOREIGN KEY (`model_id`) REFERENCES `model` (`model_id`)
);

```

### Fault report:

```

CREATE TABLE `fault_report` (
  `Id_of_report` int NOT NULL,
  `serial_number` int NOT NULL,
  `order_id` int NOT NULL,
  `problem_description` varchar(80) DEFAULT NULL,
  PRIMARY KEY (`Id_of_report`),
  KEY `Ser_number_idx` (`serial_number`),

```

```

KEY `order_id_idx` (`order_id`),
CONSTRAINT `order_id` FOREIGN KEY (`order_id`) REFERENCES `order_placed` (`order_id`) ON UPDATE CASCADE,
CONSTRAINT `Ser_number` FOREIGN KEY (`serial_number`) REFERENCES `shipment` (`Serial_number`) ON
UPDATE CASCADE
);

```

### Test:

```

CREATE TABLE `test` (
  `test_id` int NOT NULL,
  `serial_number` int NOT NULL,
  `test_date` date NOT NULL,
  `Type_of_test` varchar(45) NOT NULL,
  `test_description` varchar(45) DEFAULT NULL,
  `test_results` varchar(45) NOT NULL,
  PRIMARY KEY (`test_id`),
  KEY `ser_number_idx` (`serial_number`),
  CONSTRAINT `serial_num` FOREIGN KEY (`serial_number`) REFERENCES `shipment` (`Serial_number`) ON UPDATE
CASCADE
);

```

### Test Lists:

```

CREATE TABLE `test_lists` (
  `Test type_id` int NOT NULL,
  `Test_type` varchar(60) DEFAULT NULL,
  `test_description` varchar(180) DEFAULT NULL,
  PRIMARY KEY (`Test type_id`)
);

```

### Required\_Raw materials:

```

CREATE TABLE `required_rawmaterials` (
  `tool_id` varchar(10) NOT NULL,
  `material_name` varchar(45) NOT NULL,
  `quantity_req_tool` int NOT NULL,
  PRIMARY KEY (`tool_id`),
  KEY `moddel_id_idx` (`tool_id`),
  CONSTRAINT `tool_id` FOREIGN KEY (`tool_id`) REFERENCES `tools` (`tool_id`)
);

```

### Line schedule:

```
CREATE TABLE `line_schedule` (  
  `L_number` varchar(45) NOT NULL,  
  `prod_number` varchar(45) NOT NULL,  
  `Confirmation_Status` varchar(10) NOT NULL,  
  PRIMARY KEY (`L_number`),  
  KEY `Prod_number_idx` (`prod_number`),  
  CONSTRAINT `Prod_number` FOREIGN KEY (`prod_number`) REFERENCES `production_line` (`Prod_number`) ON  
  UPDATE CASCADE  
);
```

### Production Line:

```
CREATE TABLE `production_line` (  
  `Prod_number` varchar(45) NOT NULL,  
  `model_Id` varchar(10) NOT NULL,  
  `products_sched_units` int NOT NULL,  
  `Sched_Date` varchar(45) NOT NULL,  
  PRIMARY KEY (`Prod_number`),  
  KEY `modl_id_idx` (`model_Id`),  
  CONSTRAINT `modl_id` FOREIGN KEY (`model_Id`) REFERENCES `model` (`model_id`)  
);
```

### Order Placed:

```
CREATE TABLE `order_placed` (  
  `order_id` int NOT NULL,  
  `customer_id` varchar(10) NOT NULL,  
  `order_date` date NOT NULL,  
  PRIMARY KEY (`order_id`),  
  KEY `cust_number_idx` (`customer_id`),  
  CONSTRAINT `cust_number` FOREIGN KEY (`customer_id`) REFERENCES `customer` (`Customer_number`) ON  
  UPDATE CASCADE  
);
```

### Order details:

```
CREATE TABLE `order_details` (  
  `order_id` int NOT NULL,  
  `model_id` varchar(10) NOT NULL,  
  `unit_price` float DEFAULT NULL,  
  `quantity` int DEFAULT NULL,  
  `Total_Cost` float DEFAULT NULL,  
  PRIMARY KEY (`model_id`),  
  KEY `mdl_id_idx` (`model_id`),
```



```

KEY `ord_id_idx` (`order_id`),
CONSTRAINT `mdl_id` FOREIGN KEY (`model_id`) REFERENCES `model` (`model_id`),
CONSTRAINT `ord_id` FOREIGN KEY (`order_id`) REFERENCES `order_placed` (`order_id`) ON UPDATE CASCADE
);

```

### Order summary:

```

CREATE TABLE `order_summary` (
  `model_id` varchar(10) NOT NULL,
  `ordered_quantity` int NOT NULL,
  PRIMARY KEY (`model_id`),
  KEY `modell_id_idx` (`model_id`),
  CONSTRAINT `modell_id` FOREIGN KEY (`model_id`) REFERENCES `model` (`model_id`)
);

```

### Tool:

```

CREATE TABLE `tools` (
  `tool_id` varchar(10) NOT NULL,
  `tool_name` varchar(45) NOT NULL,
  `tool_description` varchar(500) DEFAULT NULL,
  PRIMARY KEY (`tool_id`)
);

```

### Model:

```

CREATE TABLE `model` (
  `model_id` varchar(10) NOT NULL,
  `size` varchar(45) DEFAULT NULL,
  `price` float DEFAULT NULL,
  `tool_id` varchar(10) NOT NULL,
  PRIMARY KEY (`model_id`),
  KEY `tool_id_idx` (`tool_id`)
);

```

### Employees:

```

CREATE TABLE `employees` (
  `emp_id` int NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `contact` varchar(45) DEFAULT NULL,
  `dept_numb` int NOT NULL,
  PRIMARY KEY (`emp_id`),
  KEY `dept_numb_idx` (`dept_numb`),
);

```

```
CONSTRAINT `dept_numb` FOREIGN KEY (`dept_numb`) REFERENCES `department` (`dept_number`) ON UPDATE  
CASCADE  
);
```

**Department:**

```
CREATE TABLE `department` (  
  `dept_number` int NOT NULL,  
  `dep_name` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`dept_number`)  
);
```

## 4. Data Dictionary Information

### 01.Data Dictionary for Customer:

Table	Column	Data Types	Constraints	Nullable	Description	Example
Customer	<u>Customer_number</u>	varchar (10)	PK	Not Null	Unique id for customer	C0634
Customer	customer_name	varchar (45)		Not Null	Name of a customer	Henson Hardware
Customer	customer_address	varchar (80)		Not Null	Address of a customer	227 Great Western st Leeds, Ls17 6NA
Customer	phone_number	varchar (15)		Not Null	Contact number of a customer	01132347654
Customer	email	varchar (45)		Not Null	Customer email id	suppliers@hensonhardware.com
Customer	corporate_y_or_n	varchar (5)		Not Null		
Customer	date_account_opened	date		Not Null	Account created date to order tools	1987-11-20
Customer	cp_id	int	FK	Not Null	Reference id of a contact person	

### 02.Data Dictionary for Contact Person:

Table	Column	Data Types	Constraints	Nullable	Description	Example
contact_person	<u>cp_id</u>	int	PK	Not Null	Unique id for contact_person	101
contact_person	first_name	varchar (45)		Not Null	Name of a contact_person	robbin
contact_person	last_name	varchar (45)		Not Null	Second Name of a contact_person	mathews

### 03.Data Dictionary for Order\_Placed:

Table	Column	Data Types	Constraints	Nullable	Description	Example
order_placed	<u>order_id</u>	int	PK	Not Null	Unique id for order_placed	101
order_placed	customer_id	varchar (10)	FK	Not Null	Reference id from a customer table	C0634
order_placed	order_date	date		Not Null	Date of order placed	2021-07-14

### 04.Data Dictionary for Order\_Details:

Table	Column	Data Types	Constraints	Nullable	Description	Example
order_details	<u>order_id</u>	int	FK	Not Null	Unique id for order_placed	401

order_details	<u>model_id</u>	varchar (10)	PK, FK	Not Null	Reference id from a model table	009
order_details	unit_price	float		Not Null	Price for each tool	9.25
order_details	quantity	int		Not Null	Number of tools ordered	10
order_details	Total_Cost	float		Not Null	Sum of tools price	227

#### 05.Data Dictionary for Order\_Summary:

Table	Column	Data Types	Constraints	Nullable	Description	Example
order_summary	<u>model_id</u>	varchar (10)	PK, FK	Not Null	Unique_id and reference_id from model table	009
order_summary	ordered_quantity	int		Not Null	Total ordered tools to check availability and rest sent to production	8

#### 06.Data Dictionary for Shipment:

Table	Column	Data Types	Constraints	Nullable	Description	Example
shipment	<u>Serial_number</u>	int	PK	Not Null	Unique id for tool	101
shipment	model_id	varchar (10)	FK	Not Null	Reference id from model	009
shipment	manufacture_date	date		Not Null	Manufacture date of a tool	2021-08-17
shipment	Shipment_status	Varchar (60)				

#### 07.Data Dictionary for Fault\_Report:

Table	Column	Data Types	Constraints	Nullable	Description	Example
fault_report	<u>Id_of_report</u>	int	PK	Not Null	Unique id for particular report	50001
fault_report	serial_number	int	FK	Not Null	Reference id from finished goods	22212
fault_report	order_id	int	FK	Not Null	Reference id from order placed	401
fault_report	problem_description	varchar (80)		-	Describing the problem of a tool	Wrenched or deformed

#### 08.Data Dictionary for Test Lists:

Table	Column	Data Types	Constraints	Nullable	Description	Example
test_lists	<u>Test_type_id</u>	int	PK	Not Null	Unique id for test type	301
test_lists	Test_type	varchar (60)		-	Type of test for a tool	Visual Damage

<b>test_lists</b>	Test_description	varchar (180)		-	Description of a test	-
-------------------	------------------	---------------	--	---	-----------------------	---

### 09.Data Dictionary for Test:

Table	Column	Data Types	Constraints	Nullable	Description	Example
<b>test</b>	<u>test_id</u>	int	PK	Not Null	Unique id for test	101
<b>test</b>	serial_number	int	FK	Not Null	Reference id from shipment	222212
<b>test</b>	Test_type_id	int	FK	Not Null	Reference id for a particular test	201
<b>test</b>	test_date	date		Not Null	Test date	2021-08-23
<b>test</b>	Type_of_test	varchar (45)		Not Null	Test type	Visual damage
<b>test</b>	test_description	varchar (45)		-	-	-
<b>test</b>	test_results	varchar (45)		Not Null	Results of test	test pass/fail

### 10.Data Dictionary for Tool:

Table	Column	Data Types	Constraints	Nullable	Description	Example
<b>tool</b>	<u>tool_id</u>	varchar (10)	PK	Not Null	Unique id for tool	BB04
<b>tool</b>	tool_name	varchar (45)		Not Null	Name of a tool	Brick Bolster
<b>tool</b>	tool_description	varchar (500)		-	Describing a tool	-

### 11.Data Dictionary for Model:

Table	Column	Data Types	Constraints	Nullable	Description	Example
<b>model</b>	<u>model_id</u>	varchar (10)	PK	Not Null	Unique id for a model	009
<b>model</b>	size	varchar (45)		-	Size of a model	50mm*50mm
<b>model</b>	price	float		-	Price of a each model	4.49
<b>model</b>	tool_id	varchar (10)	FK	Not Null	Reference id from a tool	BB04

### 12.Data Dictionary for Inventory\_Finished\_Goods:

Table	Column	Data Types	Constraints	Nullable	Description	Example
<b>inventory_finis hed_goods</b>	<u>serial_number</u>	Int	PK	Not Null	Unique id for each tool	222212
<b>inventory_finis hed_goods</b>	Model_id	varchar (10)		Not Null	Model of tool	009
<b>inventory_finis hed_goods</b>	manufactured_date	date		Not Null	Manudactured date	2021-10-11

### 13.Data Dictionary for Production\_Line:

Table	Column	Data Types	Constraints	Nullable	Description	Example
production_line	<u>Prod_number</u>	varchar (45)	PK	Not Null	Unique id for production line	9001A
production_line	products_sched_units	int		Not Null	Number of products to schedule	10
production_line	Sched_Date	varchar (45)		Not Null	Date to manufacture tools	2021-08-9
production_line	Model_id	varchar (10)	FK	Not Null	Reference id from a model	009

### 14.Data Dictionary for Line\_Schedule:

Table	Column	Data Types	Constraints	Nullable	Description	Example
line_schedule	<u>L_number</u>	varchar (45)	PK	Not Null	Unique id for lining products	112
line_schedule	prod_number	varchar (45)	FK	Not Null	Reference id	9001A
line_schedule	Confirmation_Status	varchar (10)		Not Null	Weather completed or pending	Completed or pending

### 15.Data Dictionary for Required\_Rawmaterials:

Table	Column	Data Types	Constraints	Nullable	Description	Example
required_rawmaterials	<u>tool_id</u>	varchar (10)	PK, FK	Not Null	Unique id	BB04
required_rawmaterials	Material_name	varchar (45)		Not Null	Name of a tool	Carbon steel
required_rawmaterials	quantity_req_tool	int		Not Null	Number of materials required	10

### 16.Data Dictionary for inventory\_raw\_materials:

Table	Column	Data Types	Constraints	Nullable	Description	Example
inventory_Raw_materials	<u>Item_number</u>	int	PK	Not Null	Unique id for tool	101
inventory_Raw_materials	item_description	varchar (45)		-	-	-
inventory_Raw_materials	quantity_oh	int		Not Null	Count of raw materials	6
inventory_Raw_materials	supplier_id	int	FK	Not Null	Reference id from supplier	201

### 17.Data Dictionary for Supplier:

Table	Column	Data Types	Constraints	Nullable	Description	Example
supplier	<u>Supplier_id</u>	int	PK	Not Null	Unique id for supplier	201
supplier	supplier_name	varchar (45)		Not Null	Name of supplier	Maison Enterprise
supplier	sup_address	varchar (80)		Not Null	-	-
supplier	phone_number	varchar (11)		Not Null	-	01135673456

### 18.Data Dictionary for raw\_material\_delivery:

Table	Column	Data Types	Constraints	Nullable	Description	Example
raw_mat_deliv ery	<u>delivery_number</u>	int	PK	Not Null	Unique id	304
raw_mat_deliv ery	Supplied_mat_numb er	int	FK	Not Null	Refrence id of a item number	144
raw_mat_deliv ery	quantity	int		Not Null	Count of material	3
raw_mat_deliv ery	Delivery_Status	varchar (10)		Not Null	Confirmation either material delivered or not	delivered or not

### 19.Data Dictionary for Employees:

Table	Column	Data Types	Constraints	Nullable	Description	Example
employees	<u>emp_id</u>	int	PK	Not Null	Unique id	101
employees	name	varchar (45)		-	-	-
employees	email	varchar (45)		-	-	-
employees	address	varchar (45)		-	-	-
employees	contact	varchar (45)		-	-	-
employees	Dept_numb	int	FK	Not Null	Reference id	202

### 20.Data Dictionary for department:

Table	Column	Data Types	Constraints	Nullable	Description	Example
-------	--------	------------	-------------	----------	-------------	---------

department	<u>Dept_number</u>	int	PK	Not Null	Unique id	202
department	Dep_name	varchar (45)			Name of department	Manufacture or testing or shipment

## 5. Demonstration of SQL Queries

The Queries and Mysql workbench screenshots are pasted for following questions.

### 01.List of tools produced by the company

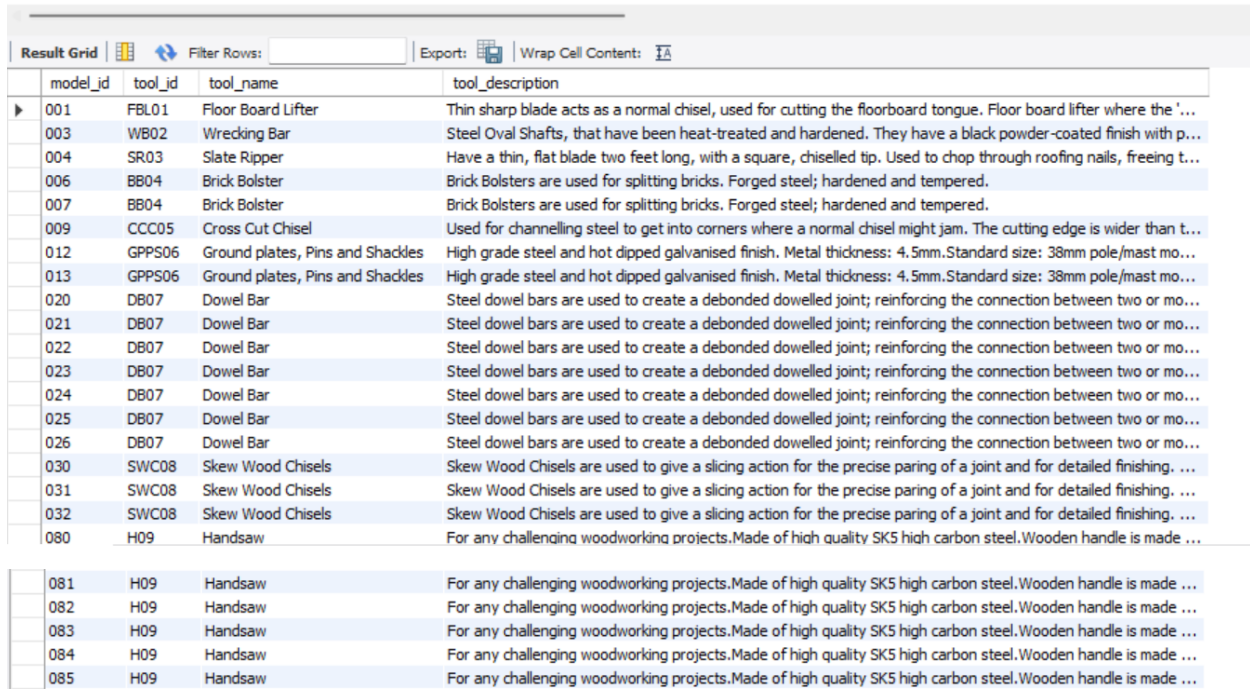
**Query:**

**Select** A.model\_id, B.tool\_id, B.tool\_name, B.tool\_description **From** model A

**Inner Join** tools B

**on** A.tool\_id = B.tool\_id

**Order By** model\_id;



	model_id	tool_id	tool_name	tool_description
▶	001	FBL01	Floor Board Lifter	Thin sharp blade acts as a normal chisel, used for cutting the floorboard tongue. Floor board lifter where the '...
	003	WB02	Wrecking Bar	Steel Oval Shafts, that have been heat-treated and hardened. They have a black powder-coated finish with p...
	004	SR03	Slate Ripper	Have a thin, flat blade two feet long, with a square, chiselled tip. Used to chop through roofing nails, freeing t...
	006	BB04	Brick Bolster	Brick Bolsters are used for splitting bricks. Forged steel; hardened and tempered.
	007	BB04	Brick Bolster	Brick Bolsters are used for splitting bricks. Forged steel; hardened and tempered.
	009	CCC05	Cross Cut Chisel	Used for channelling steel to get into corners where a normal chisel might jam. The cutting edge is wider than t...
	012	GPPS06	Ground plates, Pins and Shackles	High grade steel and hot dipped galvanised finish. Metal thickness: 4.5mm.Standard size: 38mm pole/mast mo...
	013	GPPS06	Ground plates, Pins and Shackles	High grade steel and hot dipped galvanised finish. Metal thickness: 4.5mm.Standard size: 38mm pole/mast mo...
	020	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	021	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	022	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	023	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	024	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	025	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	026	DB07	Dowel Bar	Steel dowel bars are used to create a debonded dowelled joint; reinforcing the connection between two or mo...
	030	SWC08	Skew Wood Chisels	Skew Wood Chisels are used to give a slicing action for the precise paring of a joint and for detailed finishing. ...
	031	SWC08	Skew Wood Chisels	Skew Wood Chisels are used to give a slicing action for the precise paring of a joint and for detailed finishing. ...
	032	SWC08	Skew Wood Chisels	Skew Wood Chisels are used to give a slicing action for the precise paring of a joint and for detailed finishing. ...
	080	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...
	081	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...
	082	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...
	083	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...
	084	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...
	085	H09	Handsaw	For any challenging woodworking projects.Made of high quality SK5 high carbon steel.Wooden handle is made ...



## 02.List of suppliers

Query:

**select \* from supplier;**

Result Grid				
Filter Rows:		Edit:		Export/Import:
Wrap Cell Content:				
Supplier_id	supplier_name	sup_address	phone_number	
201	Alex Raw material pvt. ltd	47 Danys Rd,Manchester M25 6TG	01135673456	
202	yashki Tool Material Suppliers	55 walkey rd, Leeds L25 0QJ	01146788988	
203	Sangsi Suppliers	666 Studfield Rd, Sheffield S10 1YH	01145468975	
204	Woodpecker Tool Traders	414 Darzil Rd, Sheffield S6 2WQ	01125679823	
205	Maison Enterprise	535 Sepcote In,Leeds L35 2JH	01167889931	
206	Salson Tools pvt. ltd	78 Attercliffe Rd, Manchester M46 7GF	01185770456	
NULL	NULL	NULL	NULL	

## 03.List of customers

Query:

**Select A.Customer\_number, A.customer\_name, A.customer\_address, A.phone\_number, A.email, A.corporate\_y\_or\_n, A.date\_account\_opened, B.first\_name, B.last\_name**  
**from customer A**  
**Left Join contact\_person B**  
**On A.cp\_id = B.cp\_id;**

Result Grid								
Filter Rows:		Export:		Wrap Cell Contents:				
Customer_number	customer_name	customer_address	phone_number	email	corporate_y_or_n	date_account_opened	first_name	last_name
C0634	Henson Hardware	227 Great Western st Leeds, Ls17 6NA	01132347654	suppliers@hensonhardware.com	Y	1987-11-20	robbin	mathews
C076	cliff's Hardware	22 Sharrow Vale Rd Manchester, M14 4LG	01142323543	Associates@cliffhardware.com	Y	1988-06-15	martin	cliff
C192	Andrea Merrick	182A Fluwood Rd,Sheffield S10 6BB	01142348765	a.merrick@CraftFurniture.co.uk	N	2001-05-01	NULL	NULL
C224	Harry William	31 hurddel Rd, Sheffield S16 PQS	01144257553	Furnicates@HarryWilliam.co.uk	Y	2012-04-15	jacob	leo
C283	Roland casey	13 Broadfield Rd, Manchester M1 2PM	01612263425	Roland@Wood-Designs.co.uk	N	2003-01-07	NULL	NULL
C296	Henson Hardware	10 Gidding Rd, Sheffield S11 8XP	01142638562	info@HensonsTools.com	N	2003-04-07	karl	johnson
C3546	Jaxon maron	35 Bansley Rd, Sheffield S11 5TS	01167423895	info@JaxonMaron.co.uk	N	2015-07-12	NULL	NULL
C421	William Oldfield	531 Richmond Rd, Sheffield S8 0QS	01142257613	W.Oldfield@CabinetMaker	N	2010-08-22	NULL	NULL
C447	George Leo	234 Flowery St, Sheffield S14 2BH	01159588195	info@GeorgeTools.com	Y	2014-01-20	william	thomas
C564	Oliver Noah	66 barley Rd, leeds L38 1HT	01157288192	OliverNoah@FurnMaker.co.uk	N	2013-07-21	NULL	NULL
C587	Lucas Arlo	128 Langsett Rd, Sheffield S10 3QS	01129069585	LucasArlo@Furnitures.co.uk	Y	2014-01-20	karl	johnson
C632	Ronnie Hunter	97 Chaple Rd, Sheffield S35 7TD	01156478996	info@RonnieMakers.co.uk	Y	2016-10-30	mick	wakefield
C639	Harrison Makers	130 QueenMary Rd, Manchester M11 4YT	01145098758	Makers@Harrison.com	Y	2014-06-09	martin	cliff
C6543	Turner Supplies	56 Garden St,Leeds LS6 4P	01132302534	info@TurnerSupplies.co.uk	Y	2012-02-03	mick	wakefield
C892	Hugo Daniel	65 Jaunty St, Leeds L12 JQH	01136782994	HugoDaniel@Furnitures.co.uk	Y	2015-09-17	martin	cliff

#### 04.Current inventory levels of finished tools

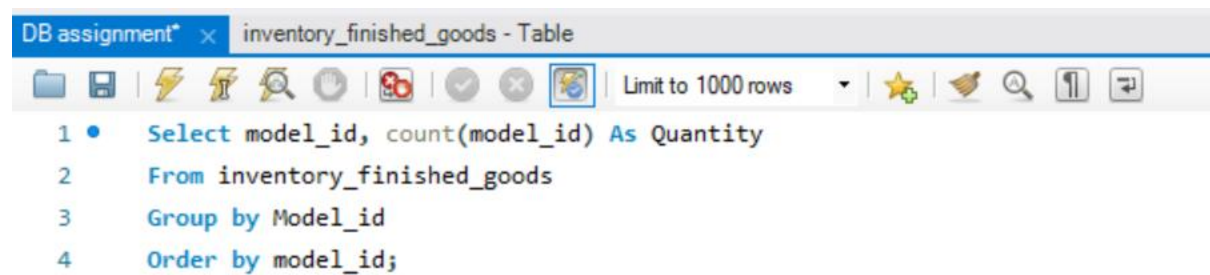
Query:

**Select** model\_id, count(model\_id) As Quantity

**From** inventory\_finished\_goods

**Group by** Model\_id

**Order by** model\_id;



The screenshot shows a database result grid window titled "Result Grid". The grid displays the results of the SQL query, with columns "model\_id" and "Quantity". The data is as follows:

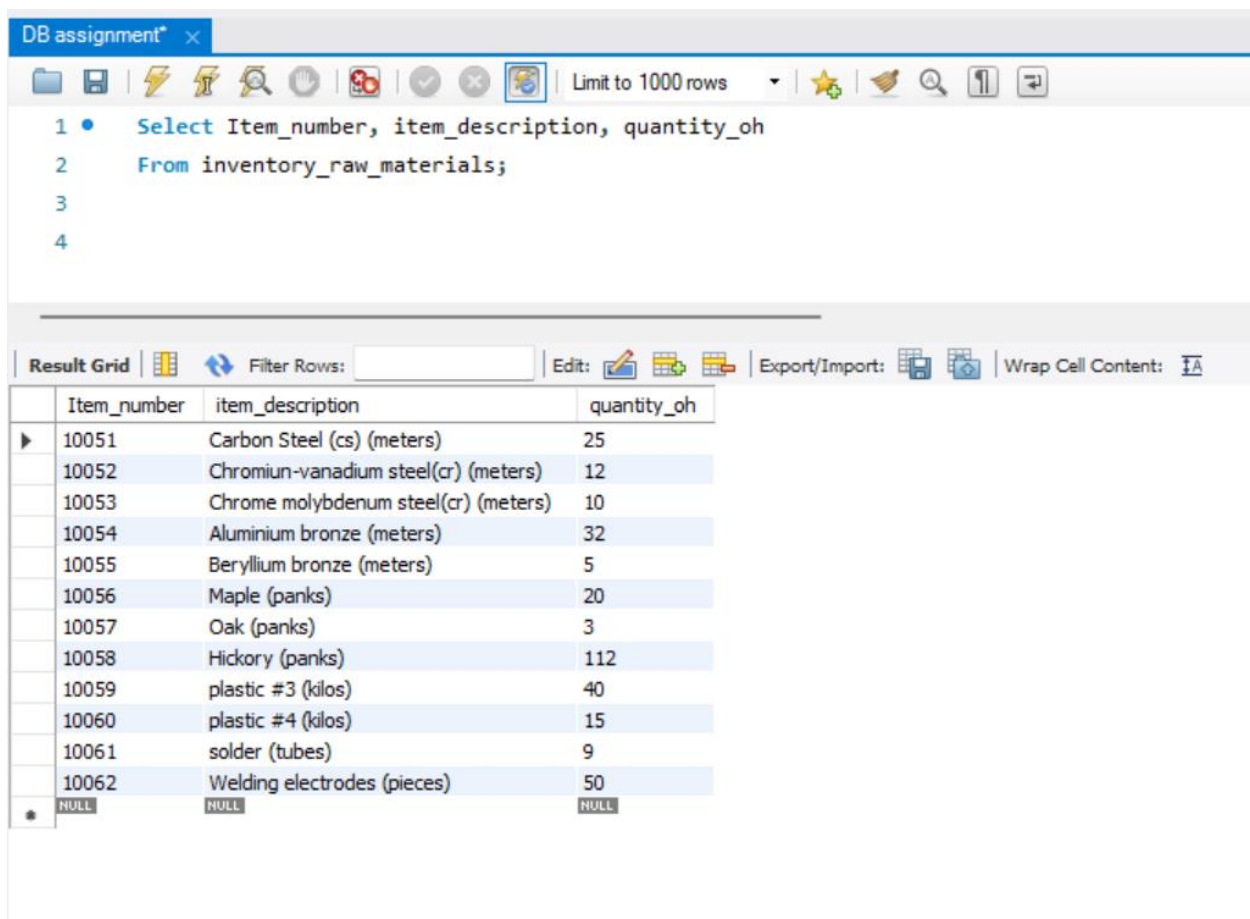
model_id	Quantity
001	6
003	9
006	8
012	4
020	3
025	5
031	8
080	3
083	4

The grid includes a toolbar with "Filter Rows", "Export", and "Wrap Cell Content" options.

## 05.Current inventory levels of raw materials

Query:

```
Select Item_number, item_description, quantity_oh  
From inventory_raw_materials;
```



The screenshot shows a database query tool interface. At the top, there's a tab labeled "DB assignment\* x". Below the tab is a toolbar with various icons for file operations, execution, and viewing. The main area displays the SQL query:

```
1 • Select Item_number, item_description, quantity_oh  
2 From inventory_raw_materials;  
3  
4
```

Below the query editor, there's a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and "Export/Import:" and "Wrap Cell Content:" options. The result grid itself is a table with the following data:

	Item_number	item_description	quantity_oh
▶	10051	Carbon Steel (cs) (meters)	25
	10052	Chromium-vanadium steel(cr) (meters)	12
	10053	Chrome molybdenum steel(cr) (meters)	10
	10054	Aluminium bronze (meters)	32
	10055	Beryllium bronze (meters)	5
	10056	Maple (panks)	20
	10057	Oak (panks)	3
	10058	Hickory (panks)	112
	10059	plastic #3 (kilos)	40
	10060	plastic #4 (kilos)	15
	10061	solder (tubes)	9
	10062	Welding electrodes (pieces)	50
*	NULL	NULL	NULL

## 06.List all the tests conducted on a given tool in the current catalogue

Query:

```
Select * From test_lists;
```

DB assignment\* x test\_lists - Table

Limit to 1000 rows

1 • `Select * from test_lists;`

---

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Test_type_id	Test_type	test_description
301	deformation	the action or process of deforming or distorting
302	visual Damage	a visual high-precision detection method for tool damage based on visual feature migration and cutting edge reconstruction is proposed
303	coating Inspection	equipment has everything you need to accurately measure; surface profile to cleanliness
304	quality control	To avoid products mistakes and produce precisely
305	pre-production anomalies	process of identifying data points that lie outside of the 'norm' and are rare in occurrence
NULL	NULL	NULL

## 07. List of raw materials required to produce a given tool

Query:

`Select * From required_rawmaterials;`

DB assignment\* x

Limit to 1000 rows

1 • `Select * from required_rawmaterials;`

---

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

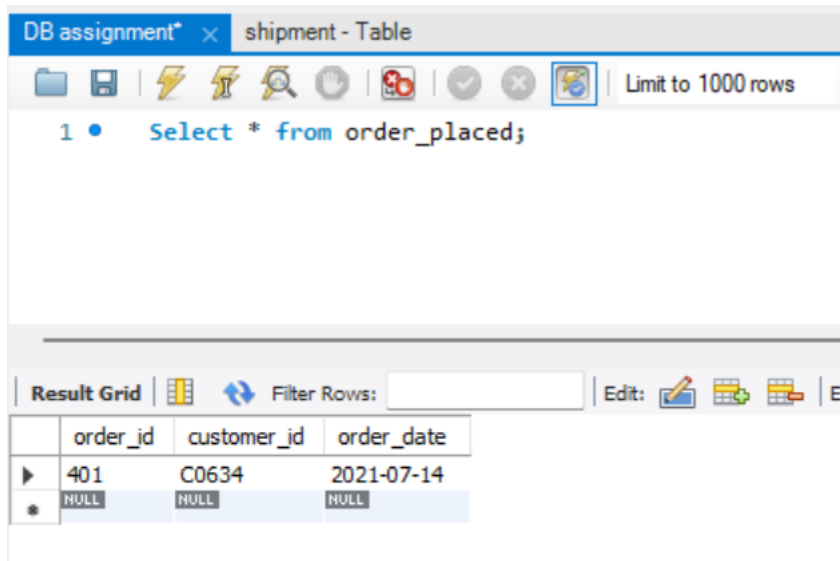
tool_id	material_name	quantity_req_tool
BB04	Aluminium Bronze	14
CCC05	Chrome vanadium steel	13
DB07	Mild Steel, Carbon Steel	10
FBL01	Carbon Steel	9
GPPS06	Carbon Steel	5
H09	Carbon Steel, rosewood, Beech	6
SR03	Chrome molybdenum steel	6
SWC08	Beryllium Bronze, Rosewood	4
WB02	Chromium-vanadium steel	7
NULL	NULL	NULL

## 08.Create a new order for an existing Customer

Query:

**Insert Into** order\_placed

**Values** (401,'C0634', 2021-07-14);



## 09.Adding a new items for a created order

Query:

**Insert Into** order\_details (order\_id, model\_id, quantity)

**Select** 401, model.model\_id,

**Case**

**When** model\_id = 003 **Then** "7"

**When** model\_id = 006 **Then** "8"

**When** model\_id = 009 **Then** "10"

**When** model\_id = 023 **Then** "9"

**Else** null

**End**

**From** model

**Where** model\_id = 003

**Or** model\_id = 006

**Or** model\_id = 009

**Or** model\_id = 023;

**Select** A.order\_id, A.model\_id, A.quantity, B.price

**From** order\_details A

**Inner Join** model B

**on** A.model\_id = B.model\_id;

DB assignment\*

Limit to 1000 rows

```

1 • Select A.order_id, A.model_id, A.quantity, B.price
2   From order_details A
3   Inner Join model B
4   on A.model_id = B.model_id;
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	order_id	model_id	quantity	price
▶	401	003	7	12
	401	006	8	5.25
	401	009	10	8.75
	401	023	9	1.5

# 10. Calculate and show the cost of each line in the order updated in question 9

Query:

Select A.order\_id, A.model\_id, A.quantity, B.price, A.quantity \* B.price as total\_price

From order\_details A

Inner Join model B

On A.model\_id = B.model\_id;

DB assignment\*

Limit to 1000 rows

```

1 • Select A.order_id, A.model_id, A.quantity, B.price, A.quantity * B.price as total_price
2   From order_details A
3   Inner Join model B
4   on A.model_id = B.model_id;
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	order_id	model_id	quantity	price	total_price
▶	401	003	7	12	84
	401	006	8	5.25	42
	401	009	10	8.75	87.5
	401	023	9	1.5	13.5

## 11. Calculate and show the total cost of the order updated in question 9

Query:

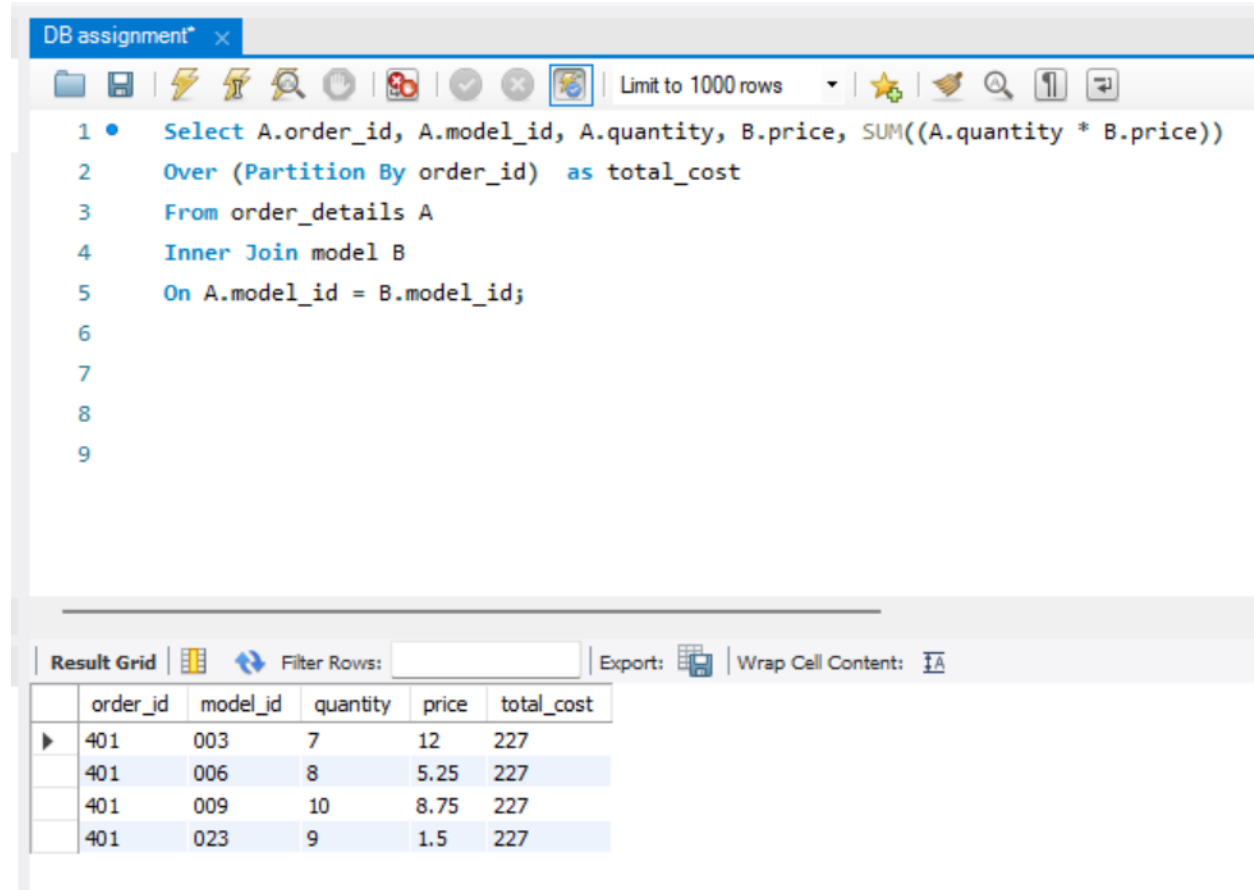
**Select** A.order\_id, A.model\_id, A.quantity, B.price, **SUM**((A.quantity \* B.price))

**Over (Partition By** order\_id) **as** total\_cost

**From** order\_details A

**Inner Join** model B

**On** A.model\_id = B.model\_id;



The screenshot shows a database query editor window titled "DB assignment". The query is as follows:

```
1 • Select A.order_id, A.model_id, A.quantity, B.price, SUM((A.quantity * B.price))
2 Over (Partition By order_id) as total_cost
3 From order_details A
4 Inner Join model B
5 On A.model_id = B.model_id;
6
7
8
9
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns: order\_id, model\_id, quantity, price, and total\_cost. The results are as follows:

order_id	model_id	quantity	price	total_cost
401	003	7	12	227
401	006	8	5.25	227
401	009	10	8.75	227
401	023	9	1.5	227

## 12. Schedule the production line for the tools requested in the order updated in question 9 that are currently out of stock

Query:

**Insert Into** production\_line (prod\_number, model\_id, products\_sched\_units, Sched\_Date)

**Select If**(A.model\_id = 009, '9001A', **If**(A.model\_id = 023, '9001B', null)), A.model\_id, A.quantity,

**Case**

**When** model\_id = 009 **Then** '2021-10-20'

**When** model\_id = 023 **Then** '2021-10-25'

**Else** null

**End**

**From** order\_details A

**Where** A.model\_id **Not In** (Select model\_id From inventory\_finished\_goods);

DB assignment\* x

Limit to 1000 rows

```
1 • select * from production_line;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Prod_number	model_Id	products_sched_units	Sched_Date
▶	9001A	009	10	2021-10-20
	9001B	023	9	2021-10-25
*	NULL	NULL	NULL	NULL

**13. For one of the tools scheduled in question 12, show the list of raw materials and quantities required for its manufacturing.**

**Query:**

**Select** A.model\_id, D.material\_name, A.products\_sched\_units \* D.quantity\_req\_tool **As** quantity

**From** production\_line A

**Join** model B

**On** A.model\_id = B.model\_id

**Join** tools C

**On** B.tool\_id = C.tool\_id

**Join** required\_rawmaterials D

**On** C.tool\_id = D.tool\_id

**Where** A.model\_id = 009;



DB assignment\*

```

1 • Select A.model_id, D.material_name, A.products_sched_units * D.quantity_req_tool As quantity
2 From production_line A
3 Join model B
4 On A.model_ID = B.model_id
5 Join tools C
6 On B.tool_id = C.tool_id
7 Join required_rawmaterials D
8 On C.tool_id = D.tool_id
9 Where A.model_id = 009;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

model_id	material_name	quantity
009	Chrome vanadium steel	130

**14. Assume raw materials are being delivered by one of the company's suppliers, create a new Delivery of raw materials order/note.**

**Query:**

**Insert Into** raw\_mat\_delivery (delivery\_number, supplied\_mat\_number, quantity, Delivery\_Status)

**Values** (3001, 10051, '10 meters', 'delivered'),

(3002, 10052, '15 meters', 'delivered'),

(3003, 10054, '5 meters', 'pending'),

(3004, 10057, '8 planks', 'pending'),

(3005, 10061, '6 tubes', 'delivered');

DB assignment\*

```

1 • Select A.delivery_number, B.supplier_id, A.Supplied_mat_number as Item_number, quantity, A.Delivery_status
2 From raw_mat_delivery A
3 Inner Join inventory_raw_materials B
4 Where A.Supplied_mat_number = B.Item_number;
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

delivery_number	supplier_id	Item_number	quantity	Delivery_status
3001	201	10051	10 meters	delivered
3002	201	10052	15 meters	delivered
3003	202	10054	5 meters	pending
3004	203	10057	8 planks	pending
3005	205	10061	6 tubes	delivered

**15. Update the inventory of raw materials by adding the raw materials delivered in question 14.**

**Query:**

**Update** inventory\_raw\_materials A

**Inner Join** raw\_mat\_delivery B

**on** A.Item\_number = B.Supplied\_mat\_number

**Set** A.quantity\_oh = A.quantity\_oh + B.quantity

**Where** B.Delivery\_Status = 'delivered';

DB assignment\* x

Limit to 1000 rows

```
1 • select * from inventory_raw_materials;
2
3
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	Item_number	item_description	quantity_oh	supplier_id
▶	10051	Carbon Steel (cs) (meters)	35	201
	10052	Chromium-vanadium steel(cr) (meters)	27	201
	10053	Chrome molybdenum steel(cr) (meters)	10	202
	10054	Aluminium bronze (meters)	32	202
	10055	Beryllium bronze (meters)	5	202
	10056	Maple (panks)	20	203
	10057	Oak (panks)	3	203
	10058	Hickory (panks)	112	203
	10059	plastic #3 (kilos)	40	204
	10060	plastic #4 (kilos)	15	204
	10061	solder (tubes)	15	205
	10062	Welding electrodes (pieces)	50	206
*	NULL	NULL	NULL	NULL

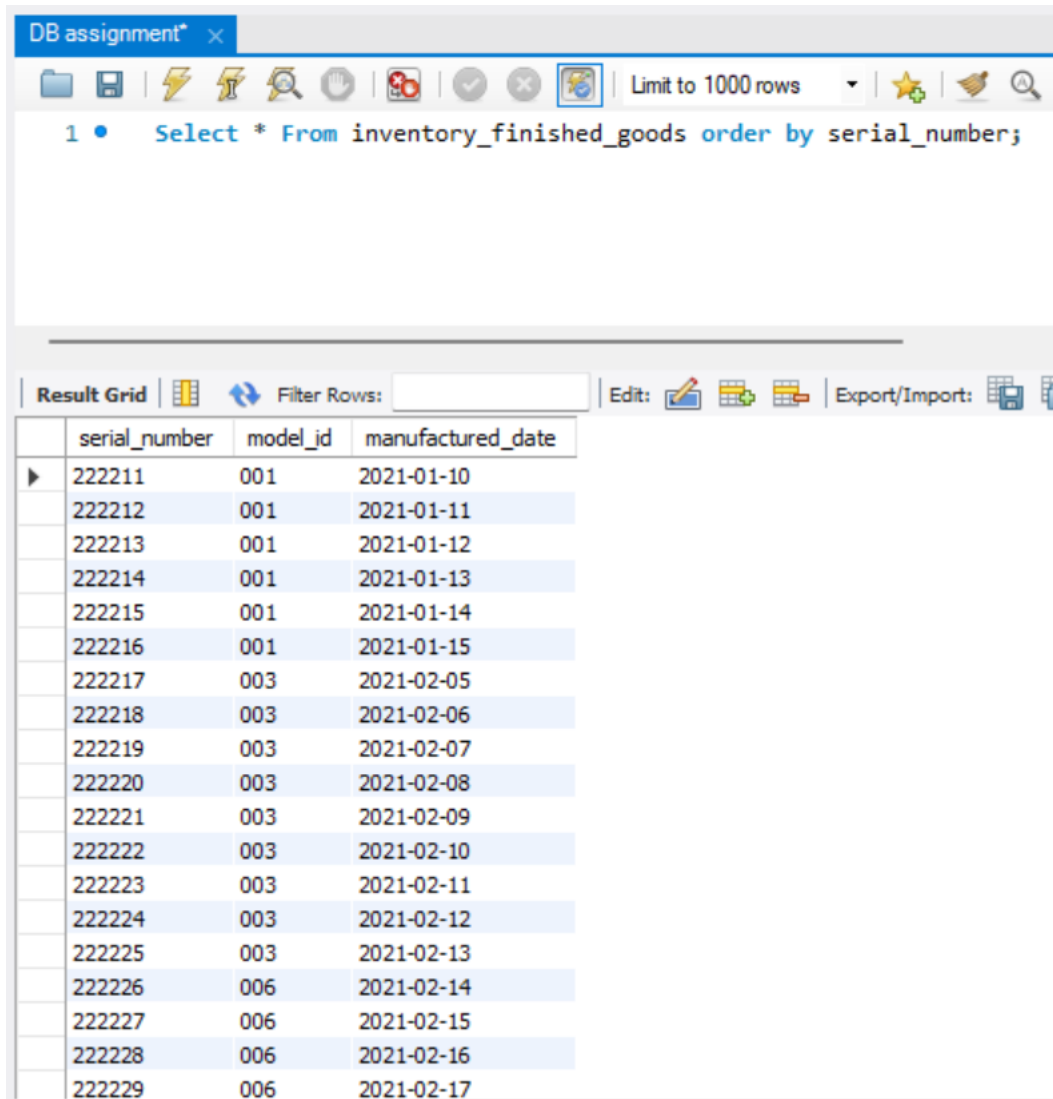
**16. Assume that the manufacturing of the tools scheduled in question 12 has finished. Add these tools to the inventory of finished tools.**

**Query:**

**Insert** Into inventory\_finished\_goods (serial\_number, model\_id, manufactured\_date)

**Values** (222261, '009', '2021-10-20'), (222262, '009', '2021-10-21'), (222263, '009', '2021-10-21'), (222264, '009', '2021-10-21'), (222265, '009', '2021-10-22'), (222266, '009', '2021-10-22'), (222267, '009', '2021-10-22'), (222268, '009', '2021-10-23'), (222269, '009', '2021-10-23'), (222270, '009', '2021-10-23'), (222271, '023', '2021-10-25'), (222272, '023', '2021-10-25'), (222273, '023', '2021-10-26'), (222274, '023', '2021-10-26'),

(222275, '023', '2021-10-26'), (222276, '023', '2021-10-27'), (222277, '023', '2021-10-27'), (222278, '023', '2021-10-27'), (222279, '023', '2021-10-27');



The screenshot shows a database query tool window titled "DB assignment\* x". The toolbar includes icons for file operations, a "Limit to 1000 rows" dropdown, and a search icon. The SQL editor contains the query: `1 • Select * From inventory_finished_goods order by serial_number;`. Below the editor is a "Result Grid" section with a "Filter Rows:" input field and "Edit:" and "Export/Import:" buttons. The result grid displays a table with three columns: `serial_number`, `model_id`, and `manufactured_date`. The data is sorted by `serial_number` in ascending order.

	serial_number	model_id	manufactured_date
▶	222211	001	2021-01-10
	222212	001	2021-01-11
	222213	001	2021-01-12
	222214	001	2021-01-13
	222215	001	2021-01-14
	222216	001	2021-01-15
	222217	003	2021-02-05
	222218	003	2021-02-06
	222219	003	2021-02-07
	222220	003	2021-02-08
	222221	003	2021-02-09
	222222	003	2021-02-10
	222223	003	2021-02-11
	222224	003	2021-02-12
	222225	003	2021-02-13
	222226	006	2021-02-14
	222227	006	2021-02-15
	222228	006	2021-02-16
	222229	006	2021-02-17

	serial_number	model_id	manufactured_date
	222230	006	2021-02-18
	222231	006	2021-02-19
	222232	006	2021-02-20
	222233	006	2021-02-21
	222234	012	2021-03-02
	222235	012	2021-03-03
	222236	012	2021-03-04
	222237	012	2021-03-05
	222238	020	2021-04-06
	222239	020	2021-04-07
	222240	020	2021-04-08
	222241	031	2021-05-10
	222242	031	2021-05-11
	222243	031	2021-05-12
	222244	031	2021-05-13
	222245	025	2021-06-15
	222246	025	2021-06-16
	222247	025	2021-06-17
	222248	025	2021-06-18

Result Grid				Filter Rows:				Edit:				Export/Import:			
	serial_number	model_id	manufactured_date												
	222249	025	2021-06-19												
	222250	031	2021-07-21												
	222251	031	2021-07-22												
	222252	031	2021-07-23												
	222253	031	2021-07-24												
	222254	080	2021-08-14												
	222255	080	2021-08-15												
	222256	080	2021-08-16												
	222257	083	2021-08-19												
	222258	083	2021-08-20												
	222259	083	2021-08-21												
	222260	083	2021-08-22												
	222261	009	2021-10-20												
	222262	009	2021-10-21												
	222263	009	2021-10-21												
	222264	009	2021-10-21												
	222265	009	2021-10-22												
	222266	009	2021-10-22												
	222267	009	2021-10-22												

222268	009	2021-10-23
222269	009	2021-10-23
222270	009	2021-10-23
222271	023	2021-10-25
222272	023	2021-10-25
222273	023	2021-10-26
222274	023	2021-10-26
222275	023	2021-10-26
222276	023	2021-10-27
222277	023	2021-10-27
222278	023	2021-10-27
222279	023	2021-10-27

**17. Using the serial number assigned to one of the tools added to the inventory of finished tools (question 16), create a shipment for the order created in question 8.**

**Query:**

**Insert Into** shipment (Serial\_number, model\_id, manufactured\_date, shipment\_status)

**Select** serial\_number, model\_id, manufactured\_date, 'Deliverd'

**From** inventory\_finished\_goods

**Where** model\_id = 009;

**Select** A.order\_id, B.Serial\_number, B.model\_id, B.manufactured\_date, B.shipment\_status

**From** order\_details A

**Inner Join** shipment B

**On** A.model\_id = B.model\_id;

DB assignment\* x shipment - Table

Limit to 1000 rows

```

1 • Select A.order_id, B.Serial_number, B.model_id, B.manufactured_date, B.shipment_status
2 From order_details A
3 Inner Join shipment B
4 on A.model_id = B.model_id;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	order_id	Serial_number	model_id	manufactured_date	shipment_status
▶	401	222261	009	2021-10-20	Deliverd
	401	222262	009	2021-10-21	Deliverd
	401	222263	009	2021-10-21	Deliverd
	401	222264	009	2021-10-21	Deliverd
	401	222265	009	2021-10-22	Deliverd
	401	222266	009	2021-10-22	Deliverd
	401	222267	009	2021-10-22	Deliverd
	401	222268	009	2021-10-23	Deliverd
	401	222269	009	2021-10-23	Deliverd
	401	222270	009	2021-10-23	Deliverd

**18. Assume one the tools added to the inventory of finished tools (question 16), has been purchased and is now returned due to a fault. Create a new Problem Report for it.**

**Query:**

**Insert Into** fault\_report (Id\_of\_report, order\_id, serial\_number, problem\_description)

**Values** (50001, 401, 222265, 'The cutting edge has deformed into different shape');

**Select** B.order\_date, A.order\_id, C.customer\_name, C.customer\_address, C.phone\_number, C.email,  
A.Id\_of\_report, A.serial\_number, A.problem\_description

**From** fault\_report A

**Inner Join** order\_placed B

**On** A.order\_id = B.order\_id

**Inner Join** customer C

**On** B.customer\_id = C.Customer\_number;

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, query execution, and navigation. The SQL editor contains the following query:

```

1 • Select B.order_date, A.order_id, C.customer_name, C.customer_address, C.phone_number, C.email, A.Id_of_report, A.serial_number, A.problem_description
2 From fault_report A
3 Inner Join order_placed B
4 On A.order_id = B.order_id
5 Inner Join customer C
6 On B.customer_id = C.Customer_number;

```

Below the query editor is the 'Result Grid' tab, which displays the results of the query in a table format. The table has 9 columns: order\_date, order\_id, customer\_name, customer\_address, phone\_number, email, Id\_of\_report, serial\_number, and problem\_description. The first row of data is as follows:

order_date	order_id	customer_name	customer_address	phone_number	email	Id_of_report	serial_number	problem_description
2021-07-14	401	Henson Hardware	227 Great Western st Leeds, Ls17 6NA	01132347654	suppliers@hensonhardware.com	50001	222265	The cutting edge has deformed into different sh...

## Conclusion:

According to my understanding of the case study, the paper demonstrates that a database was designed for the Merrill Hand Tool Company. The case study was re-evaluated in order to make changes to the Final E-R diagram for this document. The database is built with the help of the Mysql workbench application. A table was constructed for each entity, as well as create statements and updated in this presentation. We can view the data dictionary information tables for all entities, attributes with datatypes in this document. In addition, I displayed the queries and attached screenshots for given questions to the best of my ability. Finally, the provided criteria were satisfied with few assumptions, bringing the report to a close.