

NCC Level-4 Diploma in Computing

Databases



Candidates Name : Fatema Akter

ID No : 00154713

Module Title : Databases

Assignment Title : Rouse Gardens

Examination Cycle: June 2016

Candidate attempting to gain an unfair advantage or colluding in anyway whatsoever (other than on joint assignments) are liable to be disqualified. Plagiarism is an offence.

Expected candidate time allocation: 35 to 40 hours

Mark	Moderated	Final
	Mark	Mark

Marker's comment:

Moderator's comment:

Statement of Confirmation of Own Work

Programmed /qualification name: Databases

Student Declaration:

I have read and understood the NCC Education's policy on Academic Dishonesty and Plagiarism.

I can confirm the following details:

Student ID/Registration number : 00154713
Name : Fatema Akter
Center Name : Daffodil International Academy.
Module Name : Databases
Assignment Title : Rouse Gardens
Number of Words : 1,504

I confirm that this is my own and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments.

Due Date: 12/04/2016

Submitted Date: 11/04/2016

Student Signature: Fatema Akter

ACKNOWLEDGEMENT

At the beginning I would like to render thanks to the almighty Allah. And so I would wish to show my special thanks, gratitude to my teacher **Mr. Shomon Hossian** well as all other teachers. I did a great deal of research and I came to know about so many recalls and it helped to increase my knowledge.

Once more, I would wish to give thanks all of them who helped me to complete this Assignment.

Table of Contents

Introduction:	5
Task-1	6
(a) Produce Entity Relationship model:	6
Record for the entity name:	6
Entity Relationship Model:	6
(b) Show the steps normalized sample data:	7
<input type="checkbox"/> 1 st normalization:	7
<input type="checkbox"/> 2 nd normalization:	9
<input type="checkbox"/> 3 rd normalization:	10
(c) Data dictionary for entity relationship model with attributes:	11
Task-2	12
(a) Created all table using SQL:	12
(b) Enter data all the order and product:	19
(c) Enter data for the entire component:	22
(d) Enter data on staff and services:	25
(e) Query for all the service of particular customer:	29
(f) Selected query the staff assigned to a service:	30
(g) Selected query all the customer details for deliveries:	31
(h) Count queries all the service which involves gray Crowley:	31
(i) Shows query the components for the product type 'Green House':	32
(j) Update the records for 33 so that Gray Crowley:	33
(k) Update the record for staff Ben Johnson:	35
(l) Update order job center Gary Crowley involved shed construction:	36
(m) Update any session involved Anita Agneson:	37
(n) Deleted the record for Anita Agnson:	39
Task-3	41
Assessment:	41
Conclusion:	43
Bibliography	44

Introduction:

Now I am going to speak assignment **Database**, in this assignment topic the **rouse gardens**. The assignment **contains three parts**, such as **Task-1: Design, Task-2: Data Entry and Data Manipulation and Task-3: Assessment**. This assignment will enable me to show my **knowledge** and understanding of Database.

Task-1

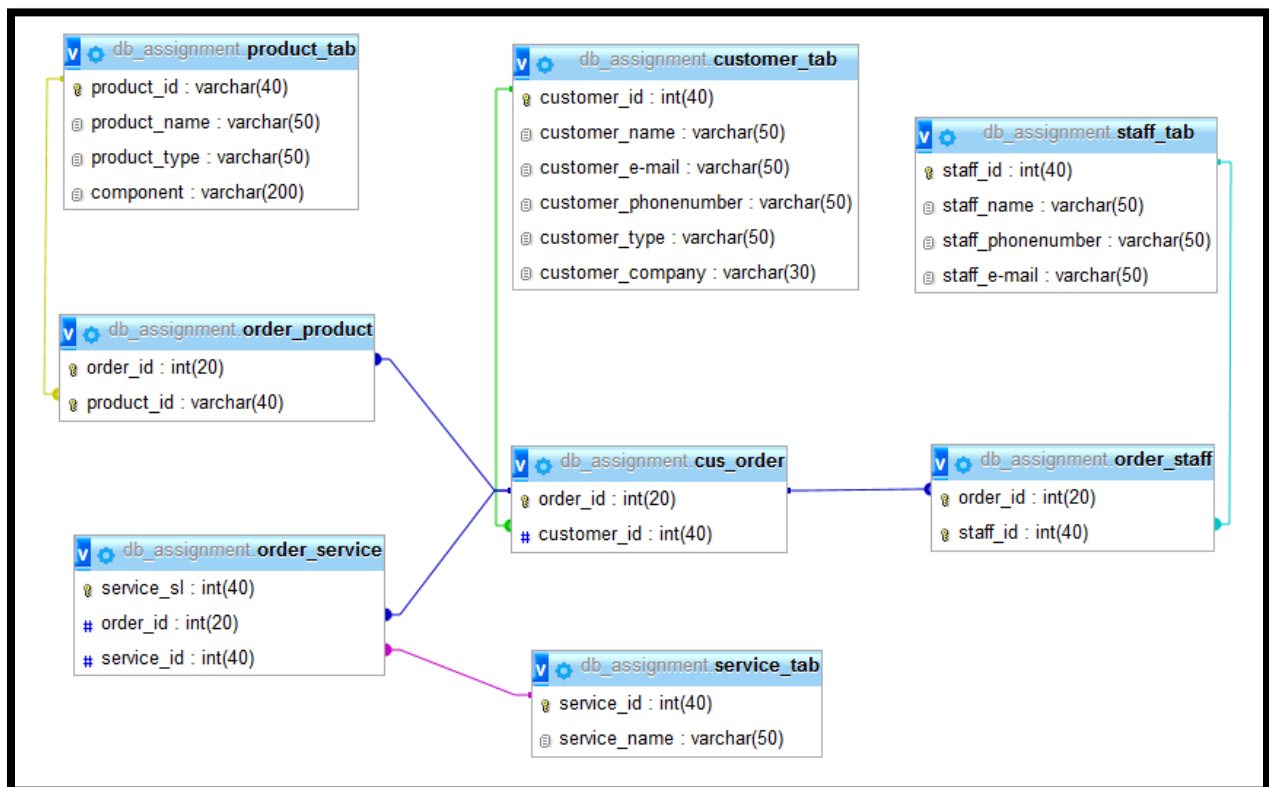
(a) Produce Entity Relationship model:

Record for the entity name:

Here I have use some entity including this:

- customer_tab
- cus_order
- order_product
- order_service
- order_staff
- product_tab
- service_tab
- staff_tab

Entity Relationship Model:



(Wesley, (2009))

Fig. 1

(b) Show the steps normalized sample data:

Normalization:

This process of **moving from data** that is not in a relational form, to a relation, and finally to a set of **ideal relation** is known **as normalization**. There are **three types** of **normalization**. Such as:


- **1st Normalization**
- **2nd Normalization and**
- **3rd Normalization form**

Described three normalization forms:

- **1st Normalization:**
 - **1st normalization** database is without **duplication data**. This assignment has **two duplication data table**.
 - These are **“order and services”** table and **“details and order”** table
 - I have **divided** into **three tables** from two table which we do not use any **duplicate data**.
 - This process name is **1st normalization**.

“Without duplicate” data table:

- Table for **“cus_order”**



	order_id	customer_id
<input type="checkbox"/> Edit Copy Delete	33	101
<input type="checkbox"/> Edit Copy Delete	11	102
<input type="checkbox"/> Edit Copy Delete	76	103

Fig. 2

- Table for “**order_service**”

+ Options			
← T →			
	service_sl	order_id	service_id
<input type="checkbox"/> Edit Copy Delete	3121	11	401
<input type="checkbox"/> Edit Copy Delete	3434	33	402
<input type="checkbox"/> Edit Copy Delete	3439	33	403
<input type="checkbox"/> Edit Copy Delete	3878	76	401
<input type="checkbox"/> Edit Copy Delete	4343	33	401
<input type="checkbox"/> Edit Copy Delete	6766	11	401

Fig. 3

Table for “**order_staff**”

+ Options		
← T →		
	order_id	staff_id
<input type="checkbox"/> Edit Copy Delete	33	104
<input type="checkbox"/> Edit Copy Delete	76	104
<input type="checkbox"/> Edit Copy Delete	11	105
<input type="checkbox"/> Edit Copy Delete	33	105
<input type="checkbox"/> Edit Copy Delete	11	106
<input type="checkbox"/> Edit Copy Delete	11	107
<input type="checkbox"/> Edit Copy Delete	33	107
<input type="checkbox"/> Edit Copy Delete	76	107

Fig. 4

- **2nd Normalization:**

- We know that where **primary key** and **foreign key** will be database system that database is a **2nd normalization**
- **Our database** is a **2nd normalization system**. Because our database **system** have **primary key** and **foreign key**.

Screen short for primary key and foreign key:

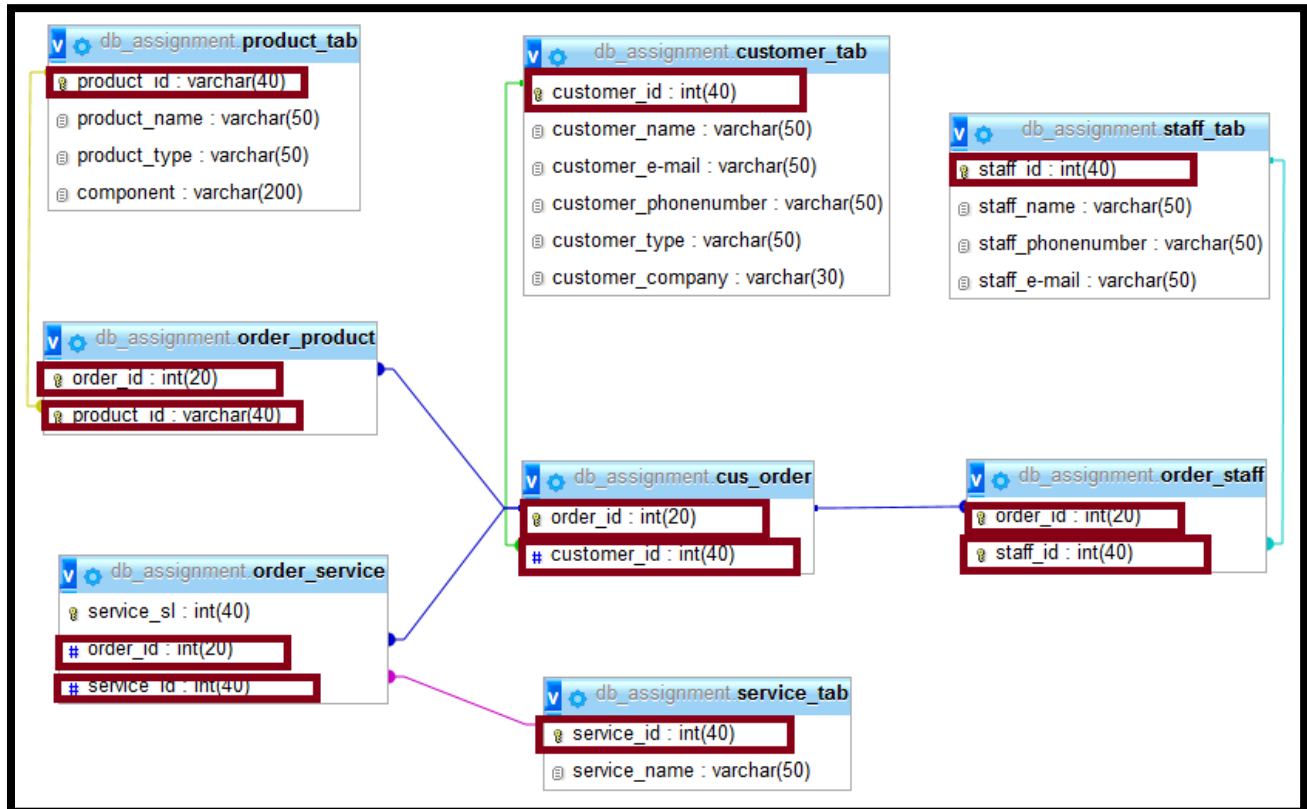


Fig. 5

- **3rd Normalization:**

- **3rd normalization** is that no **data dependency**. That means database will be without data dependency.
- Our data base is a **3rd normalization** database system because I have don't use any dependency data.

(Codd, (1970)

“No dependency” data table:

+ Options				order_id	customer id
<input type="checkbox"/>				33	101
<input type="checkbox"/>				11	102
<input type="checkbox"/>				76	103

Fig. 6

+ Options						
← T →						
	customer_id	customer name	customer_e-mail	customer_phonenumber	customer_type	customer_company
<input type="checkbox"/>	101	John Herring	PrivateIndividual@gmail.com	+19293652546	Private	Individual
<input type="checkbox"/>	102	Robinson Cars	robinson_cars@gmail.com	+19293652522	Private	Company
<input type="checkbox"/>	103	Walton Job Center	walton_job_center@gmail.com	+19293652544	Public	Organisation

Fig. 7

(c) Data dictionary for entity relationship model with attributes:

Entity Name	Attributes	Length	Key	Type
customer_tab	customer_id	40	PRIMARY KEY	int
	customer_name	50	NOT NULL	varchar
	customer_email	50	NOT NULL	varchar
	customer_phonenumber	50	NOT NULL	varchar
	customer_type	50	NOT NULL	varchar
	customer_company	50	NOT NULL	varchar
cus_order	order_id	20	PRIMARY KEY	int
	customer_id	40	FOREIGN KEY	int
order_product	order_id	20	FOREIGN KEY	int
	product_id	40	FOREIGN KEY	varchar
order_service	service_sl	40	PRIMARY KEY	int
	order_id	20	FOREIGN KEY	int
	service_id	40	FOREIGN KEY	int
order_staff	order_id	20	FOREIGN KEY	int
	staff_id	40	FOREIGN KEY	int
product_tab	product_id	40	PRIMARY KEY	varchar
	product_name	50	NOT NULL	varchar
	product_type	50	NOT NULL	varchar
	component	200	NOT NULL	varchar
service_tab	service_id	40	PRIMARY KEY	int
	service_name	50	NOT NULL	varchar
staff_tab	staff_id	40	PRIMARY KEY	int
	staff_name	50	NOT NULL	varchar
	staff_phonenumber	50	NOT NULL	varchar
	staff_email	50	NOT NULL	varchar

Fig. 8

Task-2

(a) Created all table using SQL:

- This **code** for database creation

```
CREATE DATABASE IF NOT EXISTS `db_assignment` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `db_assignment`;
```

Fig. 9

- This picture for **all database tables**

Table	Action	Rows	Type	Collation	Size	Overhead
customer_tab	Browse Structure Search Insert Empty Drop	~3	InnoDB	latin1_swedish_ci	16 KiB	-
cus_order	Browse Structure Search Insert Empty Drop	~3	InnoDB	latin1_swedish_ci	32 KiB	-
order_product	Browse Structure Search Insert Empty Drop	~4	InnoDB	latin1_swedish_ci	32 KiB	-
order_service	Browse Structure Search Insert Empty Drop	~6	InnoDB	latin1_swedish_ci	48 KiB	-
order_staff	Browse Structure Search Insert Empty Drop	~6	InnoDB	latin1_swedish_ci	32 KiB	-
product_tab	Browse Structure Search Insert Empty Drop	~4	InnoDB	latin1_swedish_ci	16 KiB	-
service_tab	Browse Structure Search Insert Empty Drop	~3	InnoDB	latin1_swedish_ci	16 KiB	-
staff_tab	Browse Structure Search Insert Empty Drop	~5	InnoDB	latin1_swedish_ci	16 KiB	-
8 tables	Sum	34	InnoDB	latin1_swedish_ci	208 KiB	0 B

Fig. 10

“customer_tab” table:

- This table for “**customer_tab**”

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	customer_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	customer_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	customer_e-mail	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	customer_phonenumber	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	customer_type	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
6	customer_company	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More

Fig. 11

- Code for “*customer_tab*” table

```
CREATE TABLE IF NOT EXISTS `customer_tab` (
  `customer_id` int(40) NOT NULL,
  `customer_name` varchar(50) NOT NULL,
  `customer_e-mail` varchar(50) NOT NULL,
  `customer_phonenumber` varchar(50) NOT NULL,
  `customer_type` varchar(50) NOT NULL,
  `customer_company` varchar(30) NOT NULL,
  PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 12

“*cus_order*” table:

- **Alter table** code for “*cus_order*”

```
ALTER TABLE `cus_order`
  ADD CONSTRAINT `cus_order_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES `customer_tab` (`customer_id`);
```

Fig. 13

- This table for “*cus_order*”

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	order_id	int(20)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	customer_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 14

- Code for “*cus_order*” table

```
CREATE TABLE IF NOT EXISTS `cus_order` (
  `order_id` int(20) NOT NULL AUTO_INCREMENT,
  `customer_id` int(40) NOT NULL,
  PRIMARY KEY (`order_id`),
  KEY `customer_id` (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=77 ;
```

Fig. 15

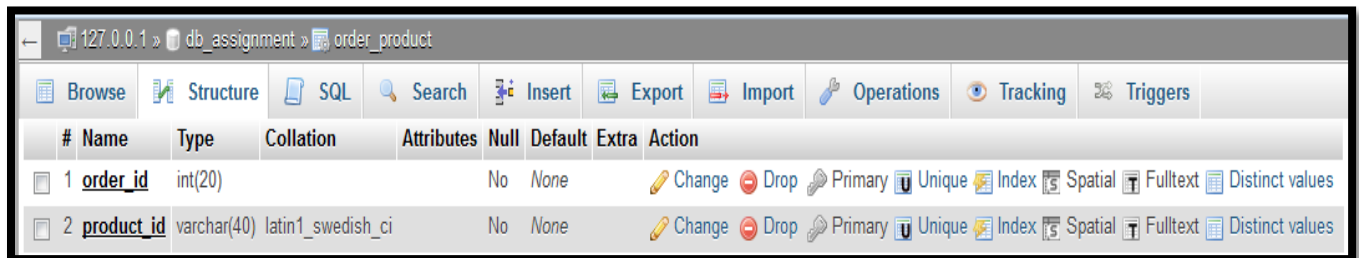
“order_product” table:

- **Alter table** code for **“order_product”**

```
ALTER TABLE `order_product`  
  ADD CONSTRAINT `order_product_ibfk_1` FOREIGN KEY (`order_id`) REFERENCES `cus_order` (`order_id`),  
  ADD CONSTRAINT `order_product_ibfk_2` FOREIGN KEY (`product_id`) REFERENCES `product_tab` (`product_id`);
```

Fig. 16

- This table for **“order_product”**



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>order_id</u>	int(20)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop
2	<u>product_id</u>	varchar(40)	latin1_swedish_ci		No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop

Fig. 17

- Code for **“order_product”** table

```
CREATE TABLE IF NOT EXISTS `order_product` (  
  `order_id` int(20) NOT NULL,  
  `product_id` varchar(40) NOT NULL,  
  PRIMARY KEY (`order_id`,`product_id`),  
  KEY `product_id` (`product_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 18

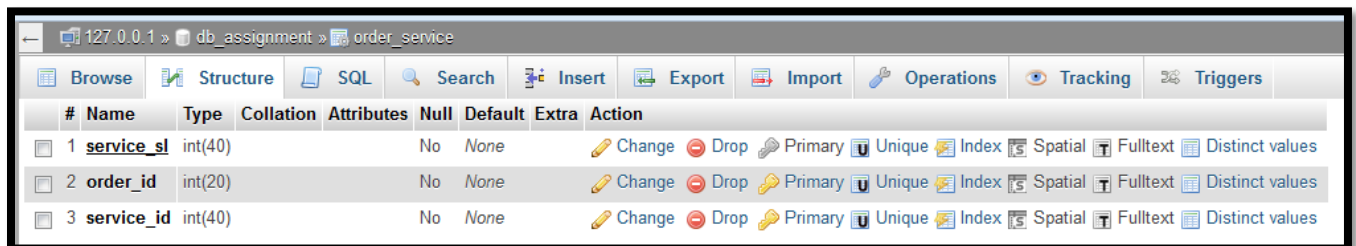
“order_service” table:

- **Alter table** code for “order_service”

```
ALTER TABLE `order_service`  
  ADD CONSTRAINT `order_service_ibfk_1` FOREIGN KEY (`order_id`) REFERENCES `cus_order` (`order_id`),  
  ADD CONSTRAINT `order_service_ibfk_2` FOREIGN KEY (`service_id`) REFERENCES `service_tab` (`service_id`);
```

Fig. 19

- This table for “order_service”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	service_sl	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	order_id	int(20)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	service_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 20

- Code for “order_service” table

```
CREATE TABLE IF NOT EXISTS `order_service` (  
  `service_sl` int(40) NOT NULL,  
  `order_id` int(20) NOT NULL,  
  `service_id` int(40) NOT NULL,  
  PRIMARY KEY (`service_sl`),  
  KEY `order_id` (`order_id`),  
  KEY `service_id` (`service_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 21

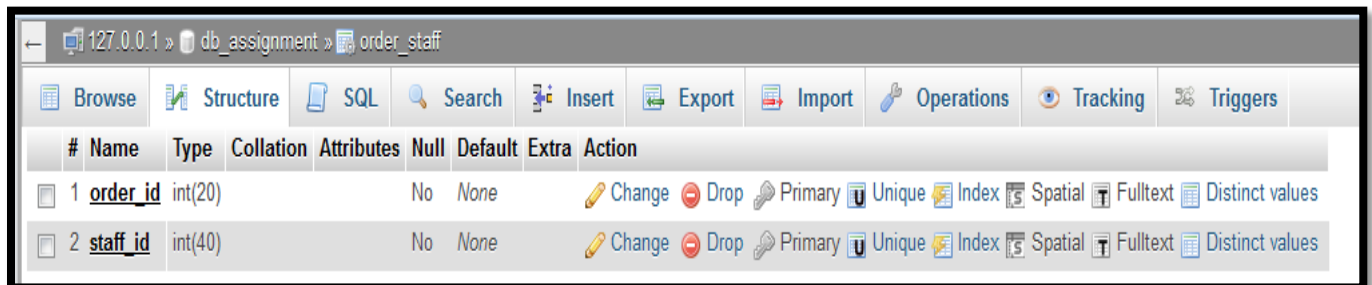
“order_staff” table:

- **Alter table** code for “order_staff”

```
ALTER TABLE `order_staff`  
  ADD CONSTRAINT `order_staff_ibfk_1` FOREIGN KEY (`order_id`) REFERENCES `cus_order` (`order_id`),  
  ADD CONSTRAINT `order_staff_ibfk_2` FOREIGN KEY (`staff_id`) REFERENCES `staff_tab` (`staff_id`);
```

Fig. 22

- This table for “order-staff”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>order_id</u>	int(20)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop
2	<u>staff_id</u>	int(40)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop

Fig. 23

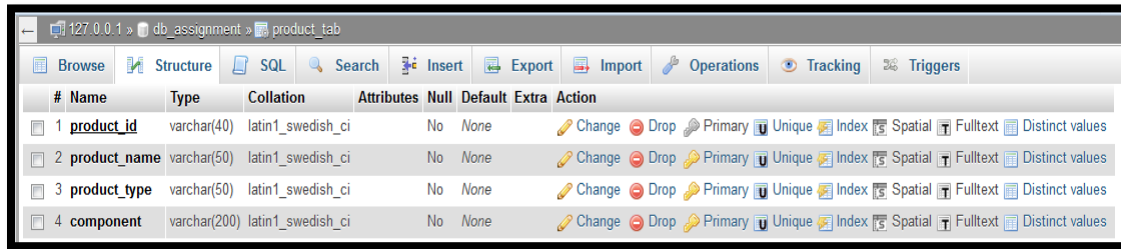
- Code for “order_staff” table

```
CREATE TABLE IF NOT EXISTS `order_staff` (  
  `order_id` int(20) NOT NULL,  
  `staff_id` int(40) NOT NULL,  
  PRIMARY KEY (`order_id`,`staff_id`),  
  KEY `staff_id` (`staff_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 24

“product_tab” table:

- This table for “*product_tab*”



The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected for the 'product_tab' table. The table has four columns: product_id, product_name, product_type, and component. Each column is of type varchar with specific lengths and collations. The product_id column is marked as the primary key.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	product_id	varchar(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	product_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	product_type	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	component	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 25

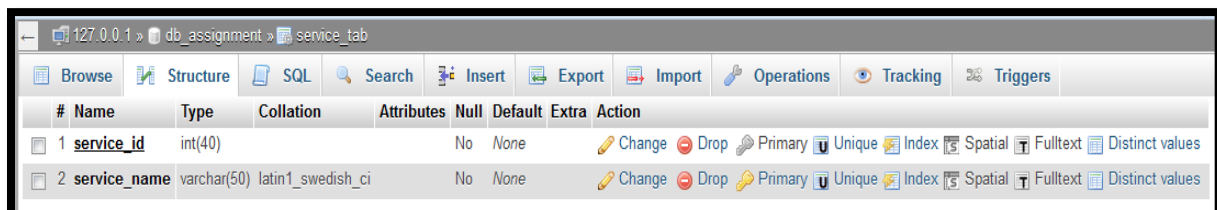
- Code for “*product_tab*” table

```
CREATE TABLE IF NOT EXISTS `product_tab` (  
  `product_id` varchar(40) NOT NULL,  
  `product_name` varchar(50) NOT NULL,  
  `product_type` varchar(50) NOT NULL,  
  `component` varchar(200) NOT NULL,  
  PRIMARY KEY (`product_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 26

“Service_tab” table:

- This table for “*service_tab*”



The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected for the 'service_tab' table. The table has two columns: service_id and service_name. service_id is of type int(40) and service_name is of type varchar(50). Both columns have a latin1_swedish_ci collation. service_id is marked as the primary key.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	service_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	service_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 27

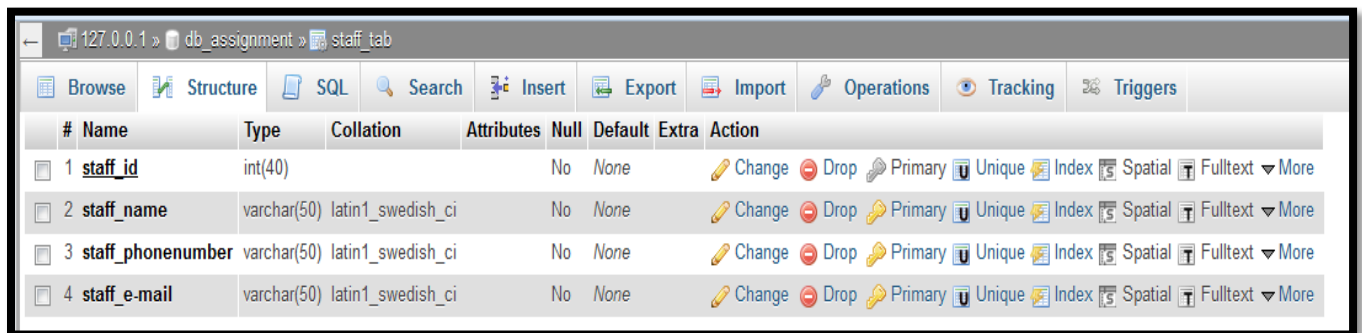
- Code for “**service_tab**” table

```
CREATE TABLE IF NOT EXISTS `service_tab` (
  `service_id` int(40) NOT NULL,
  `service_name` varchar(50) NOT NULL,
  PRIMARY KEY (`service_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Fig. 28

“Staff_tab” table:

- This table for “**staff_tab**”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	staff_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	staff_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	staff_phonenumber	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	staff_e-mail	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More

Fig. 29

- Code for “**staff_tab**” table

```
CREATE TABLE IF NOT EXISTS `staff_tab` (
  `staff_id` int(40) NOT NULL,
  `staff_name` varchar(50) NOT NULL,
  `staff_phonenumber` varchar(50) NOT NULL,
  `staff_e-mail` varchar(50) NOT NULL,
  PRIMARY KEY (`staff_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

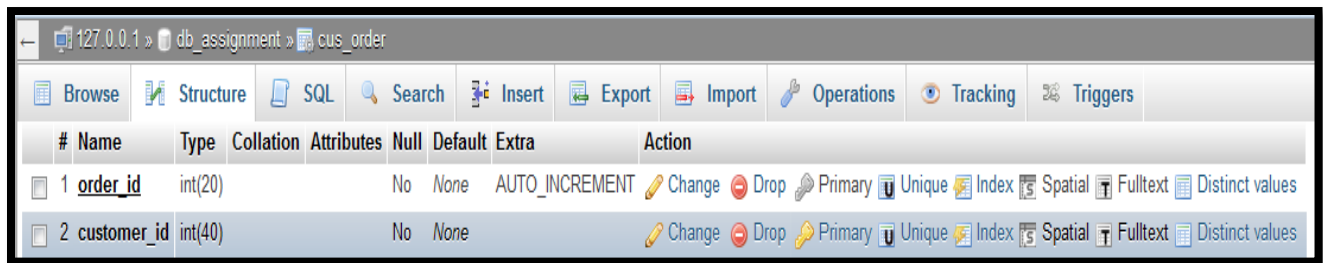
Fig. 30

(Connolly, (2004))

(b)Enter data all the order and product:

“cus_order” table:

- Table for “cus_order”

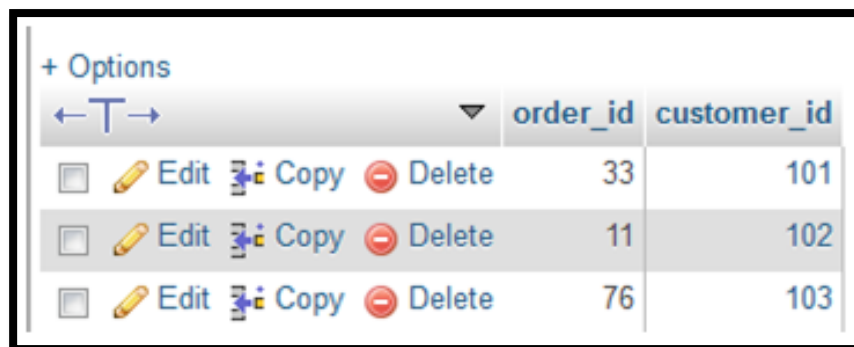


The screenshot shows the MySQL Workbench interface with the 'cus_order' table structure displayed. The table has two columns: 'order_id' and 'customer_id'. Both are of type 'int(20)' and 'int(40)' respectively, with 'None' for collation and 'None' for default. The 'order_id' column is marked as 'AUTO_INCREMENT' and 'Primary'. The 'customer_id' column is marked as 'Primary'. The 'Action' column for each row contains icons for 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', and 'Distinct values'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	order_id	int(20)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	customer_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 31

- Output for “cus_order” table



The screenshot shows the output of the 'cus_order' table. It displays a table with two columns: 'order_id' and 'customer_id'. The table contains three rows of data: (33, 101), (11, 102), and (76, 103). Each row has icons for 'Edit', 'Copy', and 'Delete'.

order_id	customer_id
33	101
11	102
76	103

Fig. 32

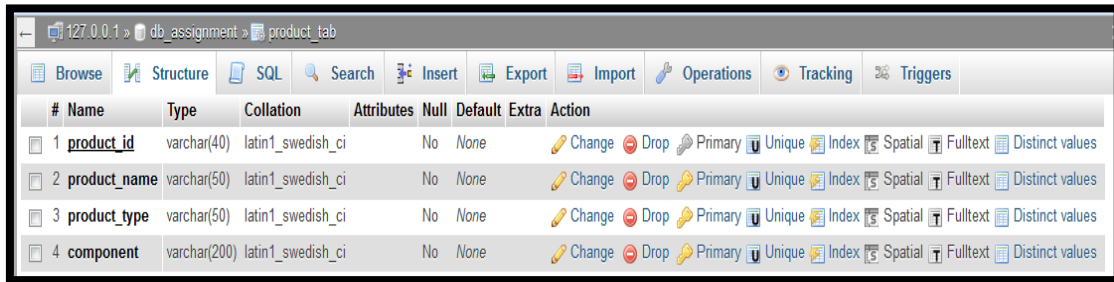
- Insert values for “cus_order”

```
INSERT INTO `cus_order` (`order_id`, `customer_id`) VALUES  
(33, 101),  
(11, 102),  
(76, 103);
```

Fig. 33

“product_tab” table:

- Table for “*product_tab*”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	product_id	varchar(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	product_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	product_type	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	component	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 34

- Output for “*product_tab*”

+ Options

















	product_id	product_name	product_type	component
  Edit  Copy  Delete A1		Small Green House	Green House	20 vertical glass panels, 2 side frames, 2 front frames, 2 roof sections, 1 door frame.
  Edit  Copy  Delete A2		Large Green House	Green House	40 vertical glass panels, 4 side frames, 2 front frames, 4 roof sections, 1 door frame
  Edit  Copy  Delete B1		Small Shed	Shed	35 Standard Timber planks. 60 Brackets, 1 door section
  Edit  Copy  Delete B2		Large Shed	Shed	60 Standard Timber planks. 120 Brackets. 1 door section

Fig. 35

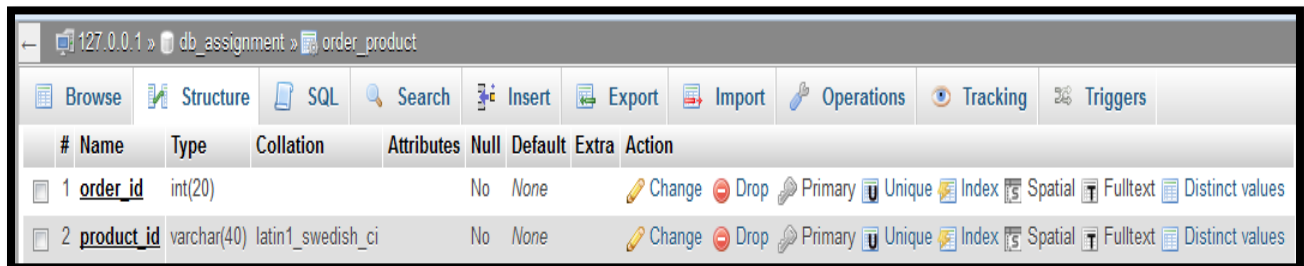
- Insert values for “*product_tab*”

```
INSERT INTO `product_tab` (`product_id`, `product_name`, `product_type`, `component`) VALUES
('A1', 'Small Green House', 'Green House', '20 vertical glass \r\npanels, 2 side \r\nframes, 2 front \r\nframes, 2 roof \r\nsections, 1 door \r\nframe. '),
('A2', 'Large Green House', 'Green House', '40 vertical glass \r\npanels, 4 side \r\nframes, 2 front \r\nframes, 4 roof \r\nsections, 1 door \r\nframe'),
('B1', 'Small Shed', 'Shed', '35 Standard Timber \r\nplanks. 60 \r\nBrackets, 1 door \r\nsection'),
('B2', 'Large Shed', 'Shed', '60 Standard Timber \r\nplanks. 120 \r\nBrackets. 1 door \r\nsection');
```

Fig. 36

“order_product” table:

- Table for “*order_product*”

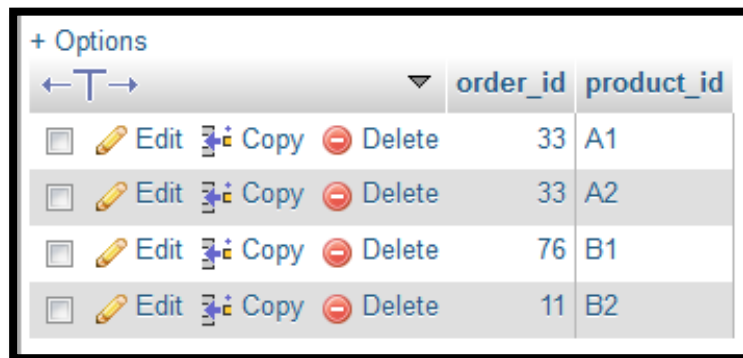


The screenshot shows the MySQL Workbench interface with the 'order_product' table selected. The table structure is displayed in a grid with columns for #, Name, Type, Collation, Attributes, Null, Default, Extra, and Action.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	<u>order_id</u>	int(20)			No	None		Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values
2	<u>product_id</u>	varchar(40)	latin1_swedish_ci		No	None		Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values

Fig. 37

- *Output* for “*order_product*”

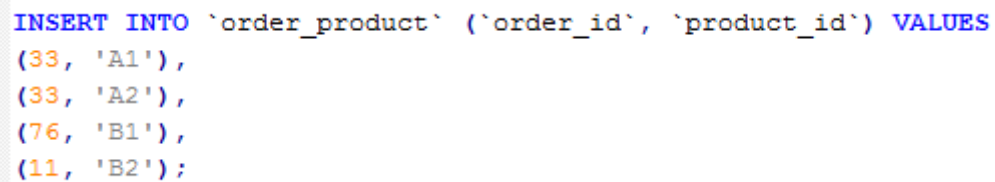


The screenshot shows the output of a query in MySQL Workbench. The table has columns 'order_id' and 'product_id'. Each row includes icons for Edit, Copy, and Delete.

	order_id	product_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	A1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	A2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	B1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	B2

Fig. 38

- *Insert values* for “*order_product*”



```
INSERT INTO `order_product` (`order_id`, `product_id`) VALUES
(33, 'A1'),
(33, 'A2'),
(76, 'B1'),
(11, 'B2');
```

Fig. 39

(c)Enter data for the entire component:
“product_tab” table:

- Table for “*product_tab*”

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	product_id	varchar(40)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	product_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	product_type	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	component	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 40

- Output for “*product_tab*”

product_id	product_name	product_type	component
A1	Small Green House	Green House	20 vertical glass panels, 2 side frames, 2 front frames, 2 roof sections, 1 door frame.
A2	Large Green House	Green House	40 vertical glass panels, 4 side frames, 2 front frames, 4 roof sections, 1 door frame
B1	Small Shed	Shed	35 Standard Timber planks. 60 Brackets, 1 door section
B2	Large Shed	Shed	60 Standard Timber planks. 120 Brackets. 1 door section

Fig. 41

- Insert values for “*product_tab*”

```
INSERT INTO 'product_tab' ('product_id', 'product_name', 'product_type', 'component') VALUES
('A1', 'Small Green House', 'Green House', '20 vertical glass \r\npanels, 2 side \r\nframes, 2 front \r\nframes, 2 roof \r\nsections, 1 door \r\nframe. '),
('A2', 'Large Green House', 'Green House', '40 vertical glass \r\npanels, 4 side \r\nframes, 2 front \r\nframes, 4 roof \r\nsections, 1 door \r\nframe'),
('B1', 'Small Shed', 'Shed', '35 Standard Timber \r\nplanks. 60 \r\nBrackets, 1 door \r\nsection'),
('B2', 'Large Shed', 'Shed', '60 Standard Timber \r\nplanks. 120 \r\nBrackets. 1 door \r\nsection');
```

Fig. 42

“customer_tab” table:

- Table for “customer_tab”

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	customer_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	customer_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	customer_e-mail	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	customer_phonenumber	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	customer_type	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
6	customer_company	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More

Fig. 43

- Output for “customer_tab”

+ Options

 		customer_id	customer_name	customer_e-mail	customer_phonenumber	customer_type	customer_company
  Edit  Copy  Delete		101	John Herring	PrivateIndividual@gmail.com	 +19293652546	Private	Individual
  Edit  Copy  Delete		102	Robinson Cars	robinson_cars@gmail.com	 +19293652522	Private	Company
  Edit  Copy  Delete		103	Walton Job Center	walton_job_center@gmail.com	 +19293652544	Public	Organisation

Fig. 44

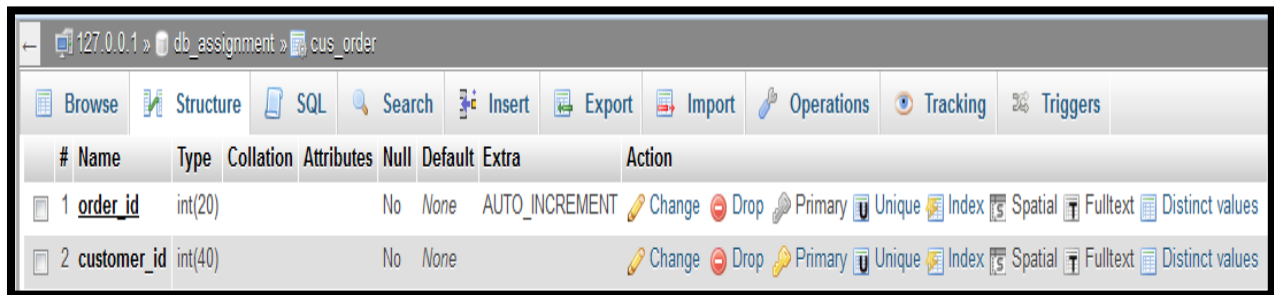
- Insert values for “customer_tab”

```
INSERT INTO `customer_tab` (`customer_id`, `customer_name`, `customer_e-mail`, `customer_phonenumber`, `customer_type`, `customer_company`) VALUES
(101, 'John Herring', 'PrivateIndividual@gmail.com', '+19293652546', 'Private', 'Individual'),
(102, 'Robinson Cars', 'robinson_cars@gmail.com', '+19293652522', 'Private', 'Company'),
(103, 'Walton Job Center', 'walton_job_center@gmail.com', '+19293652544', 'Public', 'Organisation');
```

Fig. 45

“cus_order” table:

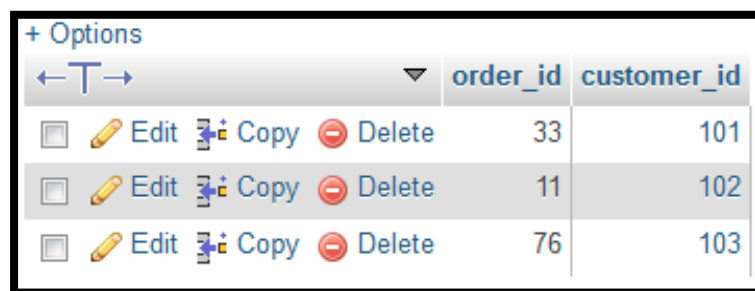
- Table for “cus_order”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	order_id	int(20)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	customer_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 46

- **Output** for “cus_order”



+ Options	
order_id	customer_id
33	101
11	102
76	103

Fig. 47

- **Insert values** for “cus_order”

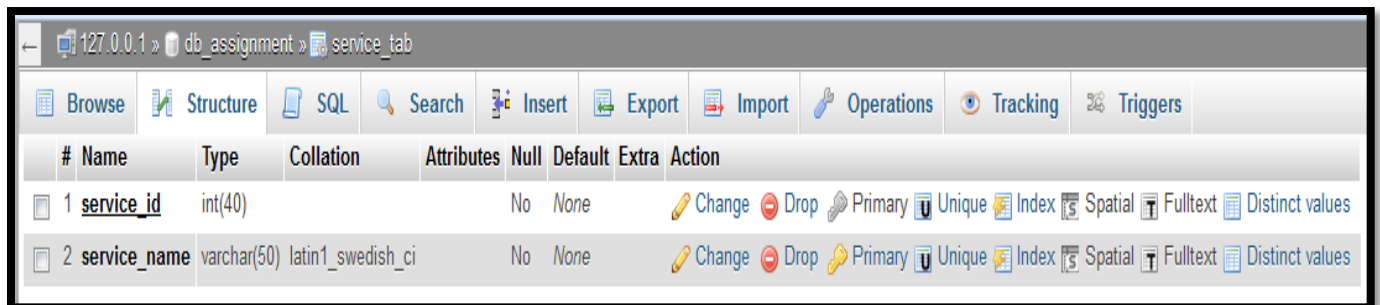
```
INSERT INTO `cus_order` (`order_id`, `customer_id`) VALUES  
(33, 101),  
(11, 102),  
(76, 103);
```

Fig. 48

(d)Enter data on staff and services:

“service_tab” table:

- Table for “**service_tab**”



The screenshot shows the MySQL Workbench interface with the 'service_tab' table structure displayed. The table has two columns: 'service_id' (int(40)) and 'service_name' (varchar(50) latin1_swedish_ci). Both columns are marked as 'No' for null, 'None' for default, and have various actions available (Change, Drop, Primary, Unique, Index, Spatial, Fulltext, Distinct values).

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	service_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	service_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 49

- Output** for “**service_tab**”



The screenshot shows the output of the 'service_tab' table. It displays three rows of data: (401, 'Delivery'), (402, 'Shed Construction'), and (403, 'Green House Construction'). Each row has icons for Edit, Copy, and Delete.

service_id	service_name
401	Delivery
402	Shed Construction
403	Green House Construction

Fig. 50

- Insert values** for “**service_tab**”

```
INSERT INTO `service_tab` (`service_id`, `service_name`) VALUES  
(401, 'Delivery'),  
(402, 'Shed Construction'),  
(403, 'Green House Construction');
```

Fig. 51

“staff_tab” table:

- Table for “**staff_tab**”

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	staff_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
2	staff_name	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	staff_phonenumber	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	staff_e-mail	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext More

Fig. 52

- Output for “**staff_tab**”

	staff_id	staff_name	staff_phonenumber	staff_e-mail
Edit Copy Delete	104	Ben Robert	+19293652588	benjohnson@gmail.com
Edit Copy Delete	105	Gary Crowley	+19293652500	grary_crowley@gmail.com
Edit Copy Delete	106	Anita Magneson	+19293652501	anita_magneson@gmail.com
Edit Copy Delete	107	Abel Mawuge	+19293652502	abel_mawuge@gmail.com
Edit Copy Delete	108	Satpal Singh	+19293652503	satpal_singh@gmail.com

Fig. 53

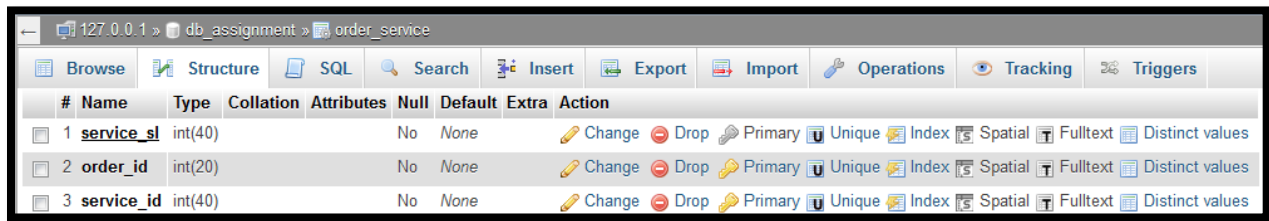
- Insert values for “**staff_tab**”

```
INSERT INTO `staff_tab` (`staff_id`, `staff_name`, `staff_phonenumber`, `staff_e-mail`) VALUES
(104, 'Ben Robert', '+19293652588', 'benjohnson@gmail.com'),
(105, 'Gary Crowley', '+19293652500', 'grary_crowley@gmail.com'),
(106, 'Anita Magneson', '+19293652501', 'anita_magneson@gmail.com'),
(107, 'Abel Mawuge', '+19293652502', 'abel_mawuge@gmail.com'),
(108, 'Satpal Singh', '+19293652503', 'satpal_singh@gmail.com');
```

Fig. 54

“order_services” table:

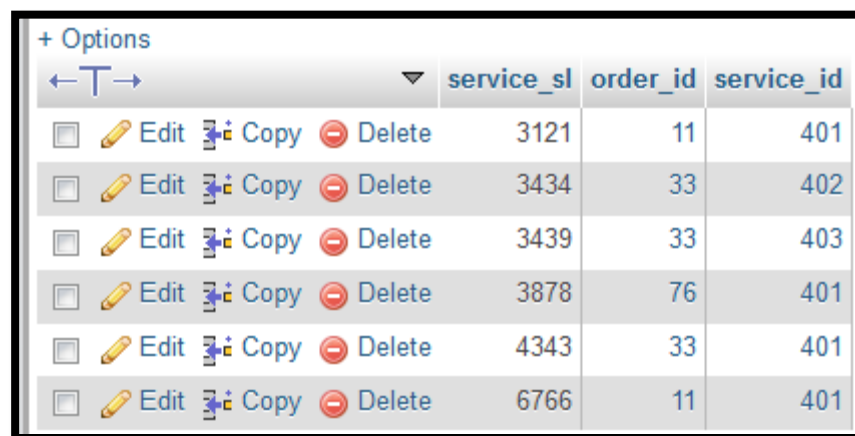
- Table for “*order_service*”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	service_sl	int(40)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop
2	order_id	int(20)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop
3	service_id	int(40)			No	None	Primary Unique Index Spatial Fulltext Distinct values	Change Drop

Fig. 55

- Output for “*order_service*”





















	service_sl	order_id	service_id
  	3121	11	401
  	3434	33	402
  	3439	33	403
  	3878	76	401
  	4343	33	401
  	6766	11	401

Fig. 56

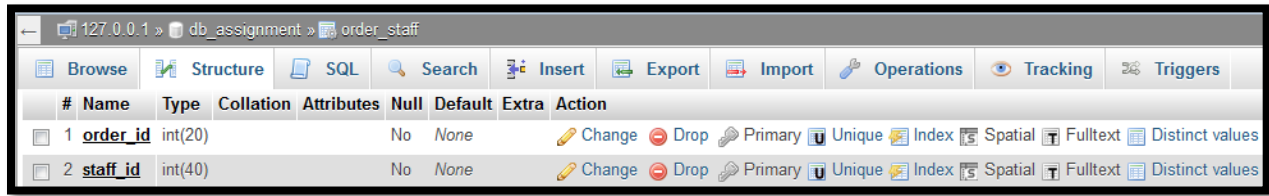
- Insert values for “*order_service*”

```
INSERT INTO `order_service` (`service_sl`, `order_id`, `service_id`) VALUES
(3121, 11, 401),
(3434, 33, 402),
(3439, 33, 403),
(3878, 76, 401),
(4343, 33, 401),
(6766, 11, 401);
```

Fig. 57

“order_staff” table:

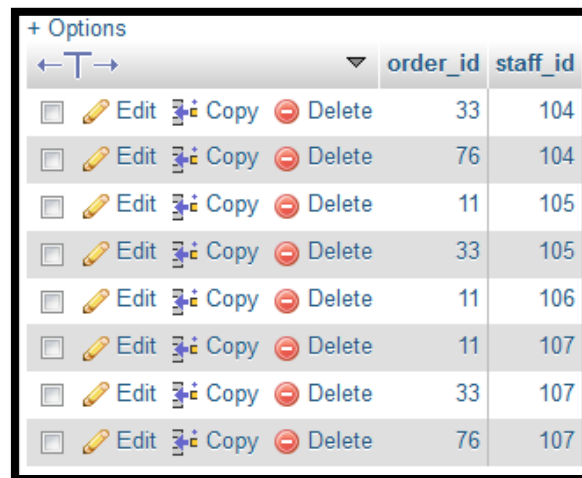
- Table for “*order_staff*”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	order_id	int(20)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	staff_id	int(40)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig. 58

- *Output* for “*order_staff*”



	order_id	staff_id
	33	104
	76	104
	11	105
	33	105
	11	106
	11	107
	33	107
	76	107

Fig. 59

- *Insert values* for “*order_staff*”

```
INSERT INTO `order_staff` (`order_id`, `staff_id`) VALUES
(33, 104),
(76, 104),
(11, 105),
(33, 105),
(11, 106),
(11, 107),
(33, 107),
(76, 107);
```

Fig. 60

(Anon., n.d.)

(e)Query for all the service of particular customer:

- Here screen short for query.

```
1 SELECT Service_id,Service_name
2 FROM service_tab st
3 WHERE st.Service_id
4 IN (SELECT DISTINCT(service_id)
5 FROM order_service os
6 WHERE os.Order_id in (SELECT Order_id
7 FROM cus_order co
8 WHERE co.Customer_id in (SELECT Customer_id
9 FROM customer tab
10 WHERE Customer_type='Private'))))
11
```

Fig. 61

- Output query

+ Options	
Service_id	Service_name
401	Delivery
402	Shed Construction
403	Green House Construction

Fig. 62

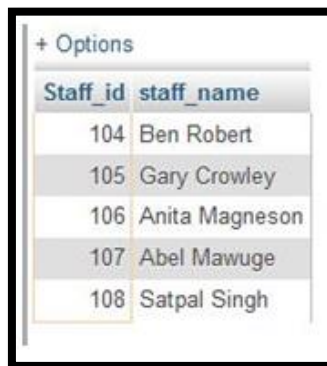
(f) Selected query the staff assigned to a service:

- Where select staff_name take from order_staff tabel

```
1 SELECT DISTINCT(st.Staff_id), staff_name
2 FROM order_staff os,staff tab st
3 WHERE os.Staff_id=st.Staff_id
4
```

Fig. 63

- Output query



The screenshot shows a database query result window titled '+ Options'. It displays a table with two columns: 'Staff_id' and 'staff_name'. The table contains five rows of data, with the last row highlighted in yellow.

Staff_id	staff_name
104	Ben Robert
105	Gary Crowley
106	Anita Magneson
107	Abel Mawuge
108	Satpal Singh

Fig. 64

(g) Selected query all the customer details for deliveries:

- Here customer_id from customer_tab table and cus_order and order_id from order_service where service_id=401

```
1 select Customer_ID, Customer_Name, Customer_Type, Customer_company
2 FROM customer_tab ct
3 WHERE ct.Customer_id IN (SELECT DISTINCT (Customer_id)
4 FROM cus_order co
5 WHERE co.Order_id in (SELECT Order_id
6 FROM order_service
7 WHERE Service_id=401))
8
9
```

Fig. 65

- Output query

+ Options			
Customer_ID	Customer_Name	Customer_Type	Customer_company
101	John Herring	Private	Individual
102	Robinson Cars	Private	Company
103	Walton Job Center	Public	Organisation

Fig. 66

(h) Count queries all the service which involves Gary Crowley:

- Query for Gary Crowley all service

```
1 SELECT count(order_id) AS services
2 FROM order_staff os
3 WHERE os.staff_id=(SELECT
4 staff_id
5 FROM staff_tab
6 WHERE staff_name='Gary Crowley')
7
```

Fig. 67

- Output query



Fig. 68

(i) Shows query the components for the product type 'Green House':

- Query code for component where product_type is "Green House"

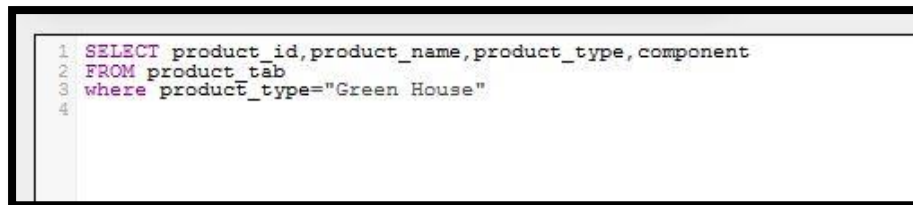


Fig. 69

- Output query

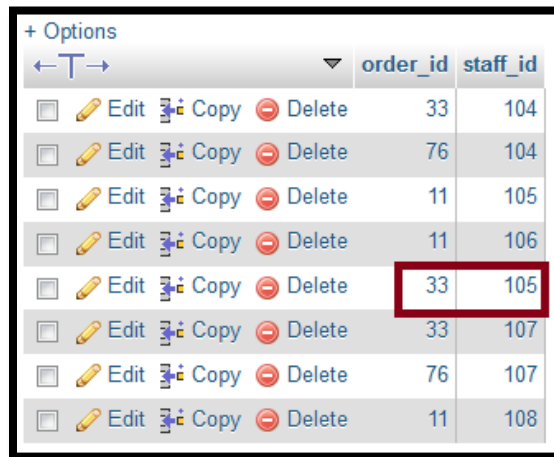
+ Options				
← T →				
	product_id	product_name	product_type	component
Edit Copy Delete A1		Small Green House	Green House	20 vertical glass panels, 2 side frames, 2 front frames, 2 roof sections, 1 door frame.
Edit Copy Delete A2		Large Green House	Green House	40 vertical glass panels, 4 side frames, 2 front frames, 4 roof sections, 1 door frame

Fig. 70

(Benyon-Davis, (2003),(2004))

(j) Update the records for 33 so that Gray Crowley:

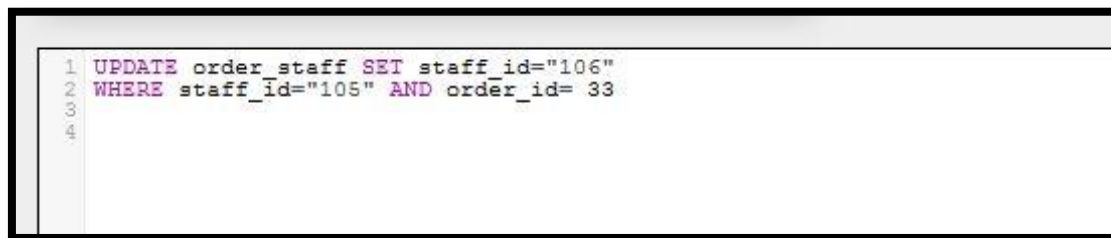
- Table for before update
- update order_staff table where staff_id 105 update will be 106 where order_id 33



	order_id	staff_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	104
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	104
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	105
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	106
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	105
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	107
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	107
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	108

Fig. 71

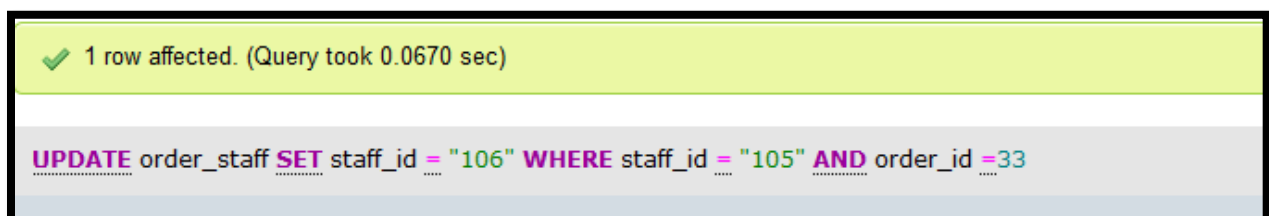
- Update query code



```
1 UPDATE order_staff SET staff_id="106"
2 WHERE staff_id="105" AND order_id= 33
3
4
```

Fig. 72

- This screen short for after update query



✓ 1 row affected. (Query took 0.0670 sec)

```
UPDATE order_staff SET staff_id = "106" WHERE staff_id = "105" AND order_id =33
```

Fig. 73

- Table for after update

























+ Options						order_id	staff_id	
← T →								
<input type="checkbox"/>		Edit		Copy		Delete	33	104
<input type="checkbox"/>		Edit		Copy		Delete	76	104
<input type="checkbox"/>		Edit		Copy		Delete	11	105
<input type="checkbox"/>		Edit		Copy		Delete	11	106
<input type="checkbox"/>		Edit		Copy		Delete	33	106
<input type="checkbox"/>		Edit		Copy		Delete	33	107
<input type="checkbox"/>		Edit		Copy		Delete	76	107
<input type="checkbox"/>		Edit		Copy		Delete	11	108

Fig. 74

(k) Update the record for staff Ben Johnson:

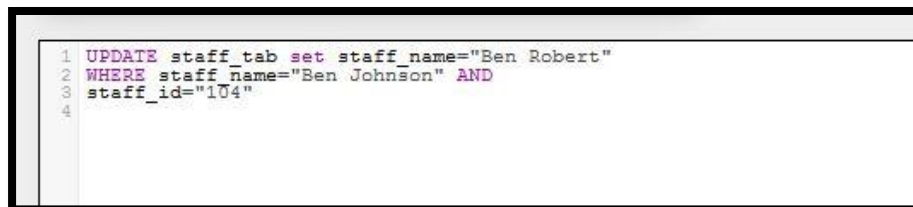
- Table for before update
- Update staff_tab table where staff_name Ben Johnson update will be Ben Robert



	staff_id	staff_name	staff_phonenumber	staff_e-mail
	104	Ben Johnson	+19293652588	benjohnson@gmail.com
	105	Gary Crowley	+19293652500	grary_crowley@gmail.com
	106	Anita Magneson	+19293652501	anita_magneson@gmail.com
	107	Abel Mawuge	+19293652502	abel_mawuge@gmail.com
	108	Satpal Singh	+19293652503	satpal_singh@gmail.com

Fig. 75

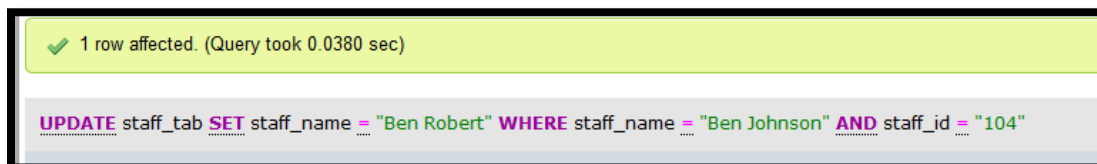
- Update query code



```
1 UPDATE staff_tab set staff_name="Ben Robert"
2 WHERE staff_name="Ben Johnson" AND
3 staff_id="104"
4
```

Fig. 76

- This screen short for after update query



✓ 1 row affected. (Query took 0.0380 sec)

```
UPDATE staff_tab SET staff_name = "Ben Robert" WHERE staff_name = "Ben Johnson" AND staff_id = "104"
```

Fig. 77

- Table for after update

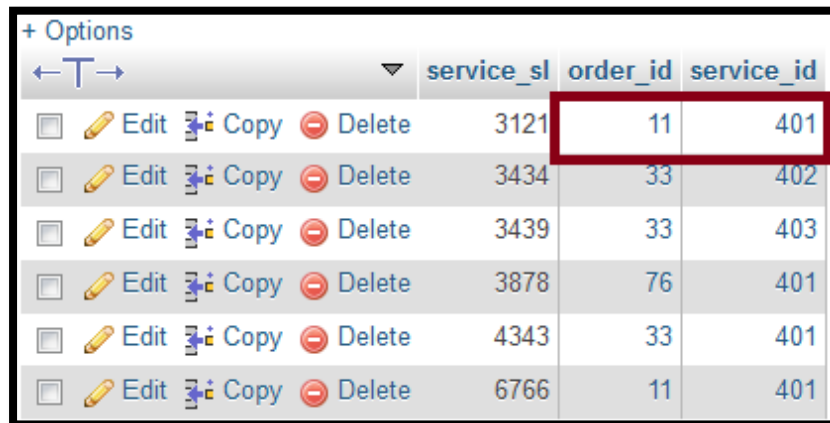


	staff_id	staff_name	staff_phonenumber	staff_e-mail
	104	Ben Robert	+19293652588	benjohnson@gmail.com
	105	Gary Crowley	+19293652500	grary_crowley@gmail.com
	106	Anita Magneson	+19293652501	anita_magneson@gmail.com
	107	Abel Mawuge	+19293652502	abel_mawuge@gmail.com
	108	Satpal Singh	+19293652503	satpal_singh@gmail.com

Fig. 78

(I) Update order job center Gary Crowley involved shed construction:

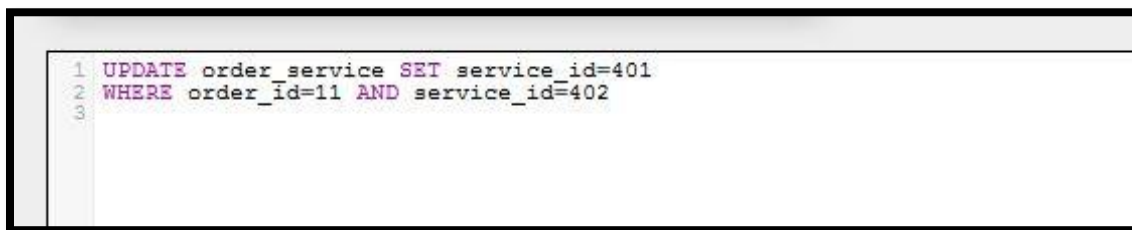
- Table for before update
- Update order_service table where service_id 401 update will be 402 where order_id 11



	service_sl	order_id	service_id
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	3121	11	401
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	3434	33	402
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	3439	33	403
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	3878	76	401
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	4343	33	401
<input type="checkbox"/> Edit <input type="image"/> Copy <input type="image"/> Delete	6766	11	401

Fig. 79

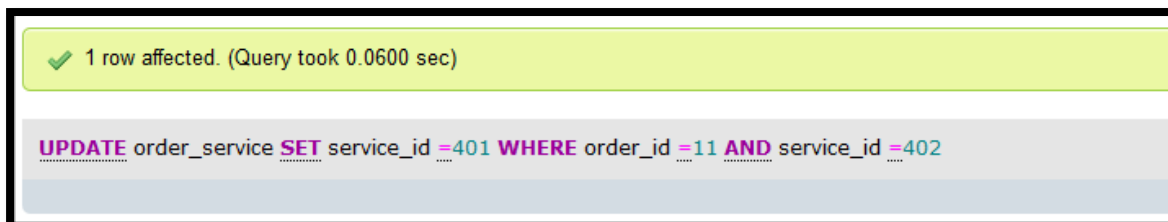
- Update query code



```
1 UPDATE order_service SET service_id=401
2 WHERE order_id=11 AND service_id=402
3
```

Fig. 80

- This screen short for after update query



✓ 1 row affected. (Query took 0.0600 sec)

UPDATE order_service SET service_id =401 WHERE order_id =11 AND service_id =402

Fig. 81

- Table for after update

+ Options				service_sl	order_id	service_id
<input type="checkbox"/>				3121	11	402
<input type="checkbox"/>				3434	33	402
<input type="checkbox"/>				3439	33	403
<input type="checkbox"/>				3878	76	401
<input type="checkbox"/>				4343	33	401
<input type="checkbox"/>				6766	11	401

Fig. 82

(m) Update any session involved Anita Agneson:

- Table for before update
- Update order_staff table where staff_id 106 update will be 108 where order_id 11

+ Options				order_id	staff_id
<input type="checkbox"/>				33	104
<input type="checkbox"/>				76	104
<input type="checkbox"/>				11	105
<input type="checkbox"/>				33	105
<input type="checkbox"/>				11	106
<input type="checkbox"/>				11	107
<input type="checkbox"/>				33	107
<input type="checkbox"/>				76	107

Fig. 83

- Update query code

```

1 UPDATE order_staff SET staff_id=106
2 WHERE order_id=11 AND staff_id=108
3 AND order_id=11

```

Fig. 84

- This screen short for successfully query update

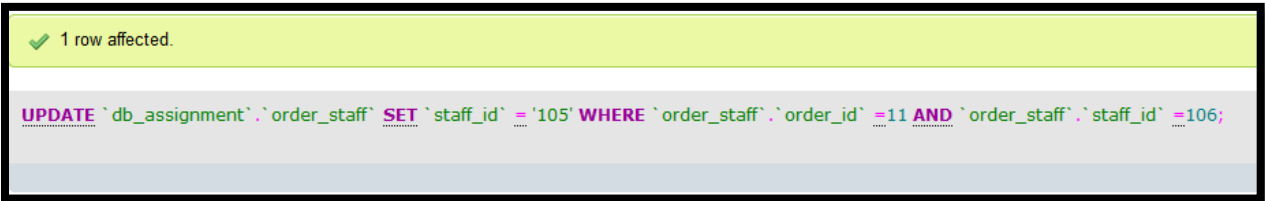


Fig. 85

- Table for after update





























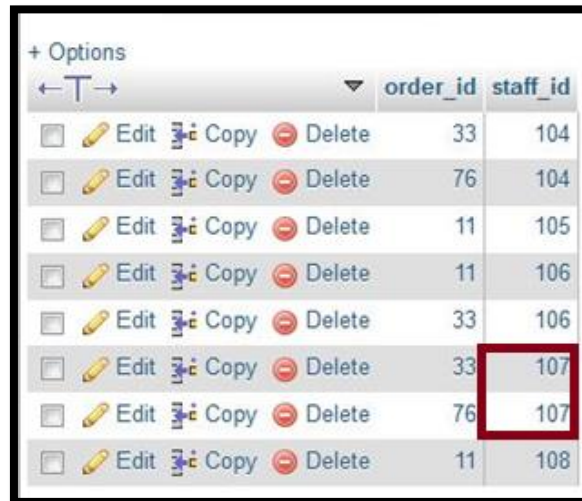
+ Options						order_id	staff_id	
  								
<input type="checkbox"/>		Edit		Copy		Delete	33	104
<input type="checkbox"/>		Edit		Copy		Delete	76	104
<input type="checkbox"/>		Edit		Copy		Delete	33	105
<input type="checkbox"/>		Edit		Copy		Delete	11	105
<input type="checkbox"/>		Edit		Copy		Delete	11	107
<input type="checkbox"/>		Edit		Copy		Delete	33	107
<input type="checkbox"/>		Edit		Copy		Delete	76	107
<input type="checkbox"/>		Edit		Copy		Delete	11	108

Fig. 86

(n)Deleted the record for Anita Agnson:

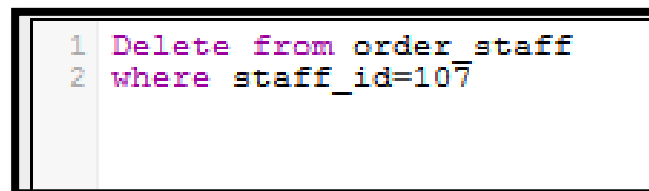
- Table for before delete
- Delete order_staff table where staff_id 107 that means Anita Agnson staff_id is 107



+ Options		
	order_id	staff_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	104
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	104
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	105
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	106
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	106
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	33	107
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	76	107
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	108

Fig. 87

- Query code for delete



```
1 Delete from order_staff
2 where staff_id=107
```

Fig. 88

- After this message box is show than click on ok for delete

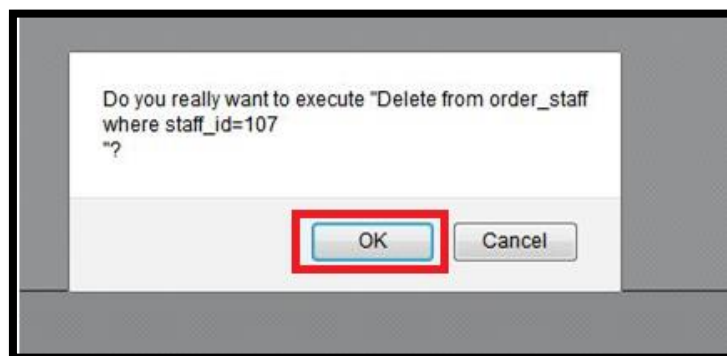


Fig. 89

- This screen short for successfully query deleted



Fig. 90

- Table for after deleted
- Here **two row** are **deleted**

+ Options

					order_id	staff_id
<input type="checkbox"/>	Edit	Copy	Delete		33	104
<input type="checkbox"/>	Edit	Copy	Delete		76	104
<input type="checkbox"/>	Edit	Copy	Delete		11	105
<input type="checkbox"/>	Edit	Copy	Delete		11	106
<input type="checkbox"/>	Edit	Copy	Delete		33	106
<input type="checkbox"/>	Edit	Copy	Delete		11	108

Fig. 91

(Hall., (2001))

Task-3

Assessment:

Requirement assessment:

- For this assignment I have used **SQL server. (My SQL version is v3.2.1).**
- I have access server used by **local host.**
- Then I have created database name. My database name is **“db_assignment”.**
- Then I have created data **8 tables** for **keeping and store data.**
- Then I have created **data relationship** between **primary and foreign key.** The data table will be **3rd normalization** from.
- Then I have completed **data dictionary** with **attributes.**
- After I have **input query.** My query run successfully.

Provides Assessment with requirement table:

Task	Requirement	Performed Task	Remark
Task 1	Entity relationship model	Analysis requirement and then Created entity relationship	done
	Normalized form	Fully used 3rd normalization form and reduce table complicity	
	Data Table	Provides data dictionary , data types, primary and foreign keys. without data redundancy	
Task 2	Table	Created database with table using SQL.	
	Enter data	Keeping and store data orders, products, component, without data lost	
		Show components	
		Input staffs and service data list which provides current and existing process	
	Query	Provides services for a particular customer which no data duplication.	
		Query which provides the staff assigned to service.	

		query which provides customer details for delivery	done
		query that count all the services which involved Gray Crowley	
		query which show the components of the product types	
	Update record Update record	update order_staff where Gray Crowley update will be delivery which order 33	
		Update where Ben Johnson update will be Ben Robert	
	Update order	Update order_service table Walton Job Center which Gary Crowley involved	
	Update session	Update session for Anita Agneson involved	
	Deleted record	Delete record for Anita Agnson	

Conclusion:

I have described all the ***process*** with ***screenshot*** which I have followed for ***design of database***. I have completed ***all the requirement in this assignment and the*** data query will be run successfully. I have ***gain a lot of knowledge*** in this assignment.

Bibliography

- Anon., n.d. *System_R/SQL_Reunion_95/*. [Online] Available at: <http://www.mcjones.org/> [Accessed 5 April 2016].
- Benyon-Davis, P..T.&.B.C., (2003),(2004). *Database Systems , Database*. 3rd ed. Palgrave Macmillan. Chapters 11, 12 & 13.Connolly.
- Codd, E.F., (1970. *A Relational Model for Large Shared Data Banks*. Retrieved on 20/05/11 from.
- Connolly, T.&.B.C., (2004). *Practical Approach to Design, Implementation, and Management*. 4th ed. Database SystemsA.
- Hall., P., (2001). *Understanding Relational Database Query Languages,Chapter 5*. 1st ed.
- Wesley, A., (2009). *Chapter 1 of Connolly, Database Systems: A Practical Approach to Design*. 5th ed.