NCC Level-4 Diploma in Computing

Databases

Candidates Name : Fatema Akter

ID No : 00154713

Title : Databases

Examination Cycle : March 2015

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 Mark	Final 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 Institute of Information Technology.

Module Name : Databases

Assignment title : Garcia Plumbing

Number of Words : 913

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:

Student Signature:

Submitted Date:

Table of Contents

Introduction:

This assignment has four parts such as TASK 1, TASK 2, TASK 3 and TASK 4. All parts of this assignment are done according to the Garcia plumbing scenario as shown in the assignment. Here I am describing about my assignment.

Task-1

Initial Design

(1.1.1)Part of Design:

Record for the name of entity:

- customer
- fitting
- job
- jobType
- jobFitting
- staff
- jobStaff
- supplier
- property

(1.1.2)Diagram of relationship Entity:

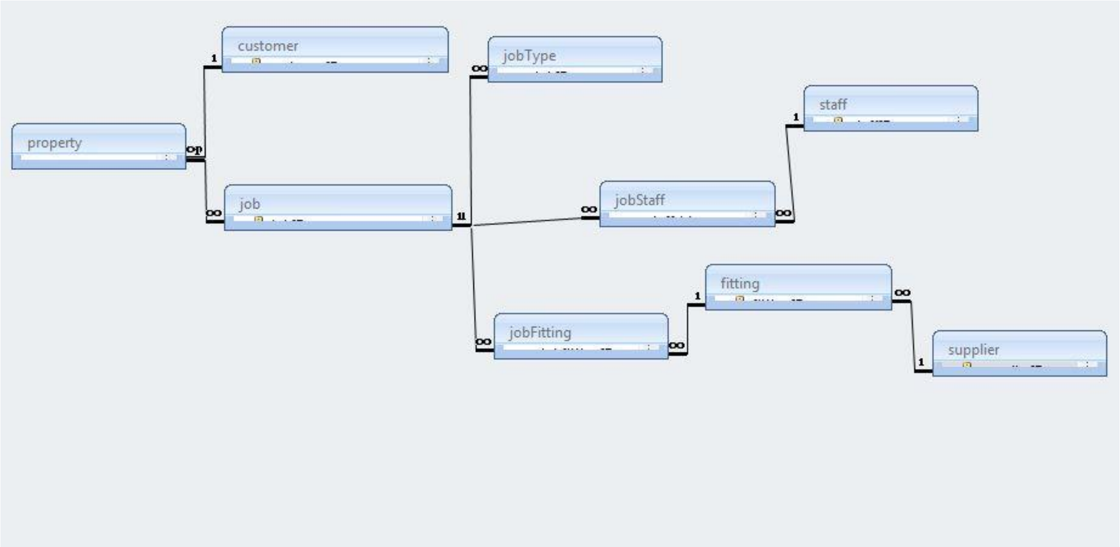


Figure no: 1.1: Diagram of relationship Entity

(1.1.3)Entity relationship:

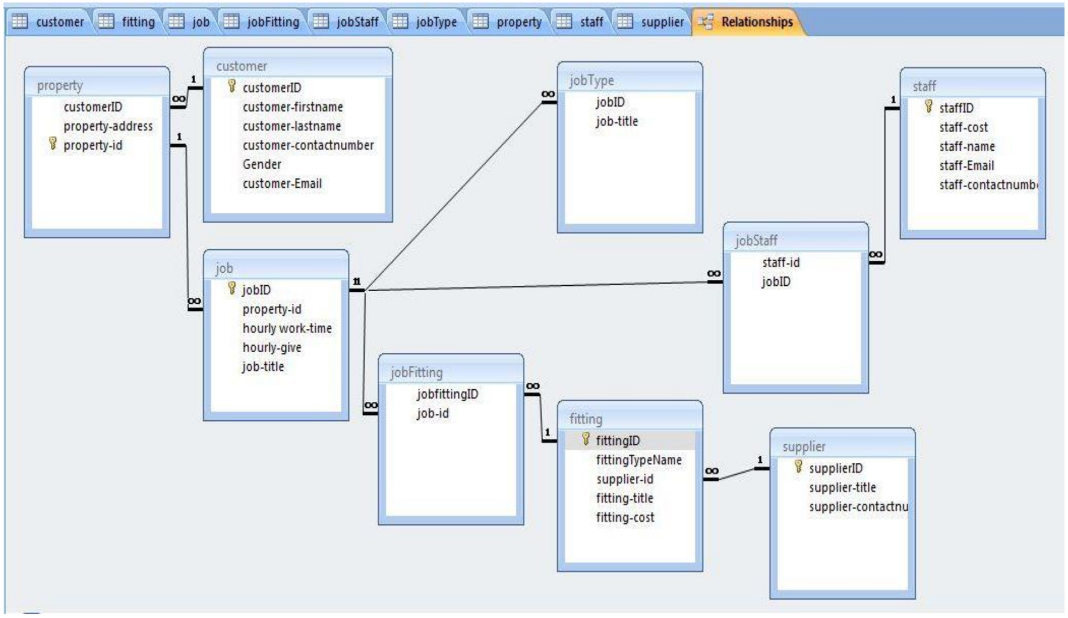
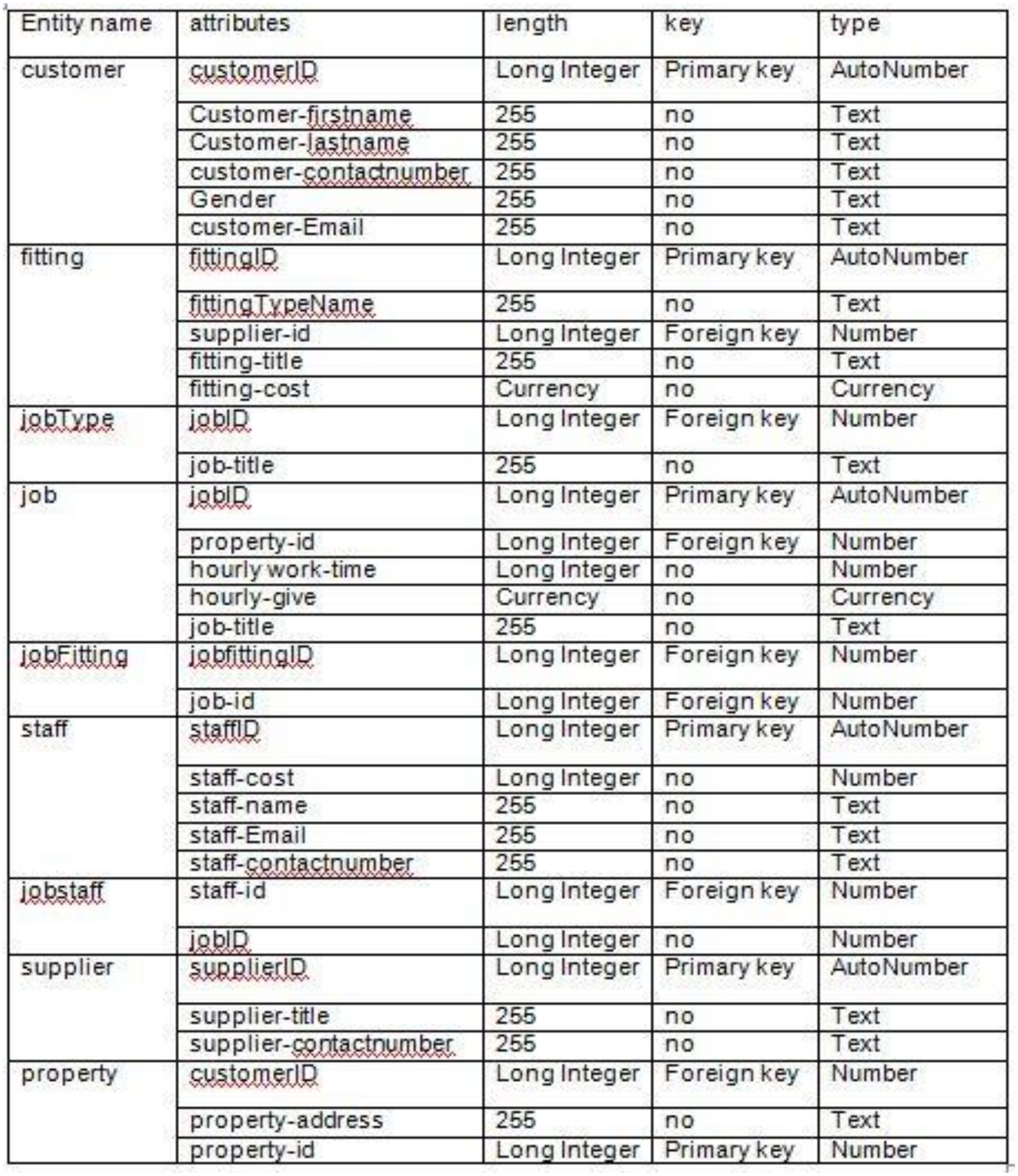


Figure no: 1.2: Diagram of Entity relationship model

(1.2)Data Dictionary



Entity name	attributes	length	key	type
customer	customerID	Long Integer	Primary key	AutoNumber
	Customer-firstname	255	no	Text
	Customer-lastname	255	no	Text
	customer-contactnumber	255	no	Text
	Gender	255	no	Text
	customer-Email	255	no	Text
fitting	fittingID	Long Integer	Primary key	AutoNumber
	fittingTypeName	255	no	Text
	supplier-id	Long Integer	Foreign key	Number
	fitting-title	255	no	Text
	fitting-cost	Currency	no	Currency
jobType	jobID	Long Integer	Foreign key	Number
	job-title	255	no	Text
job	jobID	Long Integer	Primary key	AutoNumber
	property-id	Long Integer	Foreign key	Number
	hourly work-time	Long Integer	no	Number
	hourly-give	Currency	no	Currency
	job-title	255	no	Text
jobFitting	jobfittingID	Long Integer	Foreign key	Number
	job-id	Long Integer	Foreign key	Number
staff	staffID	Long Integer	Primary key	AutoNumber
	staff-cost	Long Integer	no	Number
	staff-name	255	no	Text
	staff-Email	255	no	Text
	staff-contactnumber	255	no	Text
jobstaff	staff-id	Long Integer	Foreign key	Number
	jobID	Long Integer	no	Number
supplier	supplierID	Long Integer	Primary key	AutoNumber
	supplier-title	255	no	Text
	supplier-contactnumber	255	no	Text
property	customerID	Long Integer	Foreign key	Number
	property-address	255	no	Text
	property-id	Long Integer	Primary key	Number

Figure no: 1.3: Display of Data Dictionary

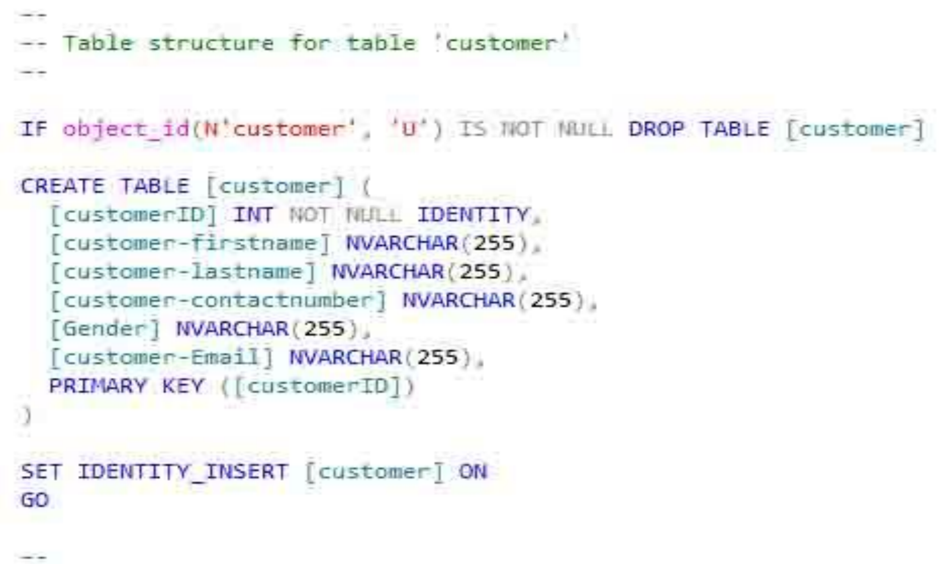
Task-2

Data and queries:

(2.1) Implementation and Data

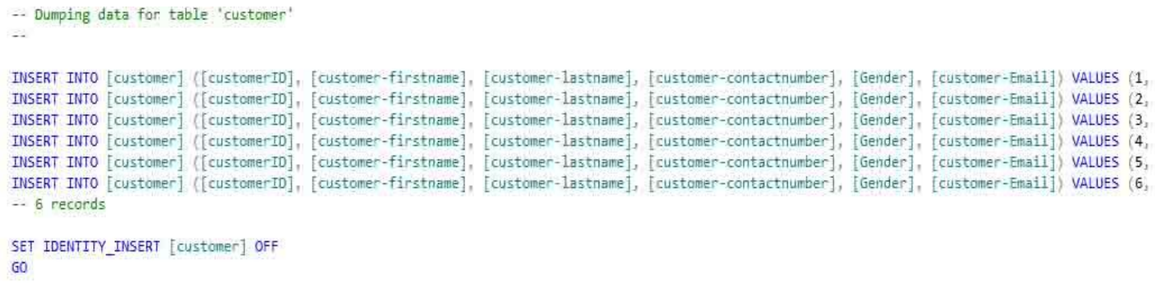
(2.1.1) Creating customer table

SQL command for “customer” table



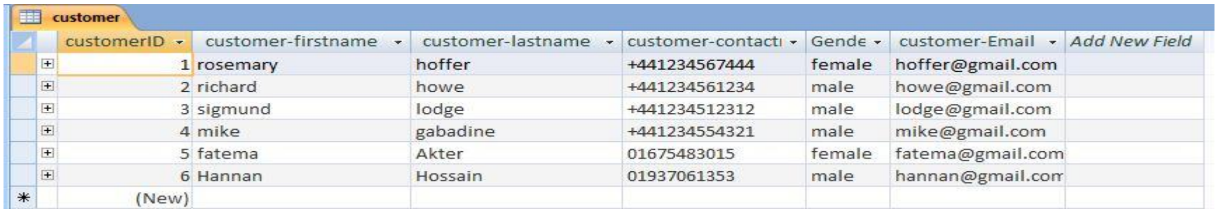
```
--
-- Table structure for table 'customer'
--

IF object_id(N'customer', 'U') IS NOT NULL DROP TABLE [customer]

CREATE TABLE [customer] (
  [customerID] INT NOT NULL IDENTITY,
  [customer-firstname] NVARCHAR(255),
  [customer-lastname] NVARCHAR(255),
  [customer-contactnumber] NVARCHAR(255),
  [Gender] NVARCHAR(255),
  [customer-Email] NVARCHAR(255),
  PRIMARY KEY ([customerID])
)

SET IDENTITY_INSERT [customer] ON
GO

--
```

Value for customer table



```
-- Dumping data for table 'customer'
--

INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (1,
INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (2,
INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (3,
INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (4,
INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (5,
INSERT INTO [customer] ([customerID], [customer-firstname], [customer-lastname], [customer-contactnumber], [Gender], [customer-Email]) VALUES (6,
-- 6 records

SET IDENTITY_INSERT [customer] OFF
GO
```

Output Result:

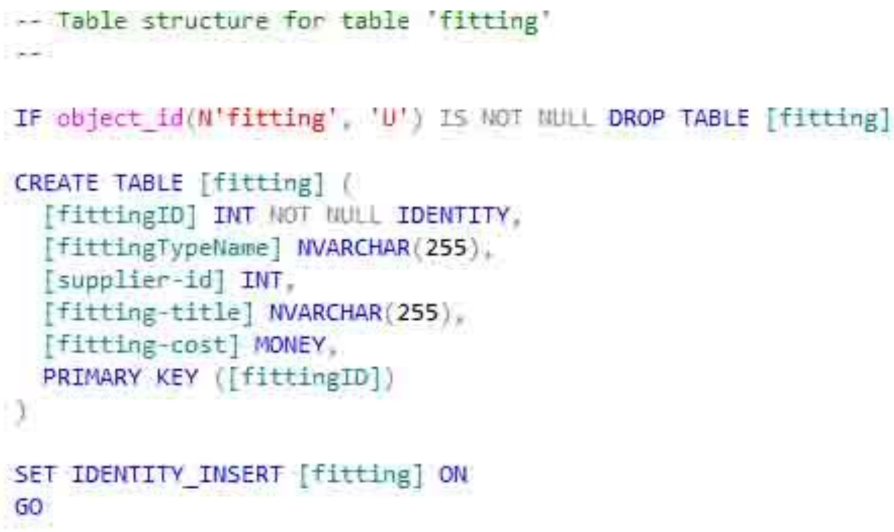


customerID	customer-firstname	customer-lastname	customer-contacti	Gende	customer-Email	Add New Field
1	rosemary	hoffer	+441234567444	female	hoffer@gmail.com	
2	richard	howe	+441234561234	male	howe@gmail.com	
3	sigmund	lodge	+441234512312	male	lodge@gmail.com	
4	mike	gabadine	+441234554321	male	mike@gmail.com	
5	fatema	Akter	01675483015	female	fatema@gmail.com	
6	Hannan	Hossain	01937061353	male	hannan@gmail.con	
(New)						

Figure no: 2.1: Display of customer table

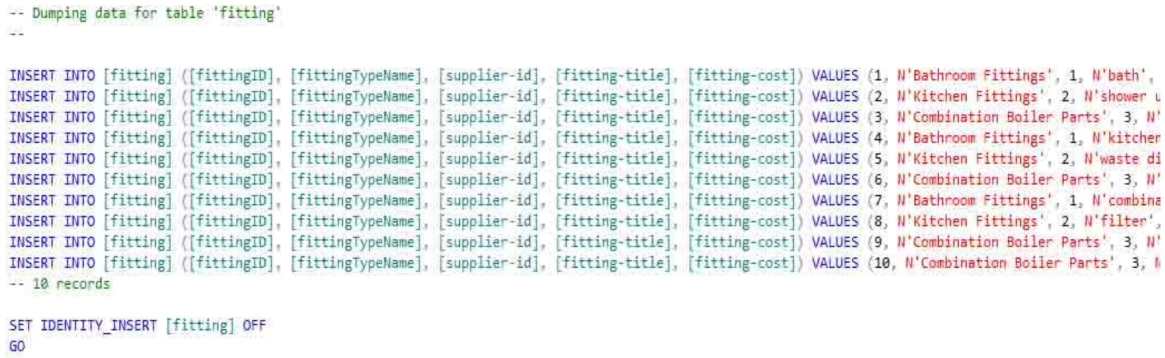
(2.1.2) Creating fitting table

SQL command for “fitting” table



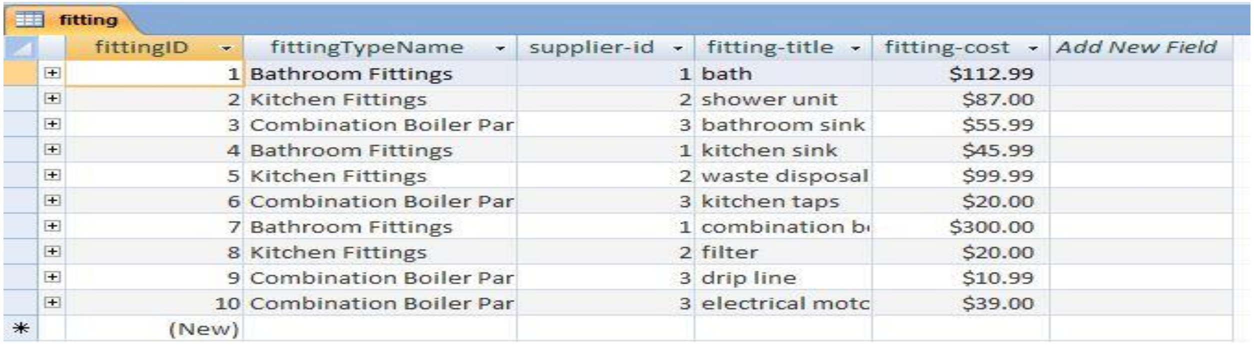
```
-- Table structure for table 'fitting'
--

IF object_id(N'fitting', 'U') IS NOT NULL DROP TABLE [fitting]

CREATE TABLE [fitting] (
  [fittingID] INT NOT NULL IDENTITY,
  [fittingTypeName] NVARCHAR(255),
  [supplier-id] INT,
  [fitting-title] NVARCHAR(255),
  [fitting-cost] MONEY,
  PRIMARY KEY ([fittingID])
)

SET IDENTITY_INSERT [fitting] ON
GO
```

Value for fitting table



```
-- Dumping data for table 'fitting'
--

INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (1, N'Bathroom Fittings', 1, N'bath',
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (2, N'Kitchen Fittings', 2, N'shower u
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (3, N'Combination Boiler Parts', 3, N'
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (4, N'Bathroom Fittings', 1, N'kitcher
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (5, N'Kitchen Fittings', 2, N'waste di
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (6, N'Combination Boiler Parts', 3, N'
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (7, N'Bathroom Fittings', 1, N'combina
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (8, N'Kitchen Fittings', 2, N'filter',
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (9, N'Combination Boiler Parts', 3, N'
INSERT INTO [fitting] ([fittingID], [fittingTypeName], [supplier-id], [fitting-title], [fitting-cost]) VALUES (10, N'Combination Boiler Parts', 3, N
-- 10 records

SET IDENTITY_INSERT [fitting] OFF
GO
```

Output Result:



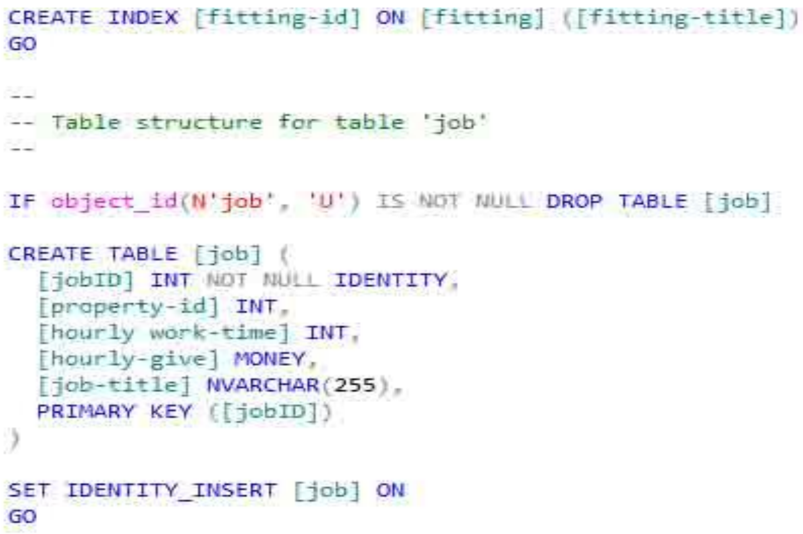
	fittingID	fittingTypeName	supplier-id	fitting-title	fitting-cost	Add New Field
⊞	1	Bathroom Fittings	1	bath	$112.99	
⊞	2	Kitchen Fittings	2	shower unit	$87.00	
⊞	3	Combination Boiler Par	3	bathroom sink	$55.99	
⊞	4	Bathroom Fittings	1	kitchen sink	$45.99	
⊞	5	Kitchen Fittings	2	waste disposal	$99.99	
⊞	6	Combination Boiler Par	3	kitchen taps	$20.00	
⊞	7	Bathroom Fittings	1	combination b	$300.00	
⊞	8	Kitchen Fittings	2	filter	$20.00	
⊞	9	Combination Boiler Par	3	drip line	$10.99	
⊞	10	Combination Boiler Par	3	electrical motc	$39.00	
*	(New)					

Figure no: 2.2: Display of fitting table

(Anon., n.d.)

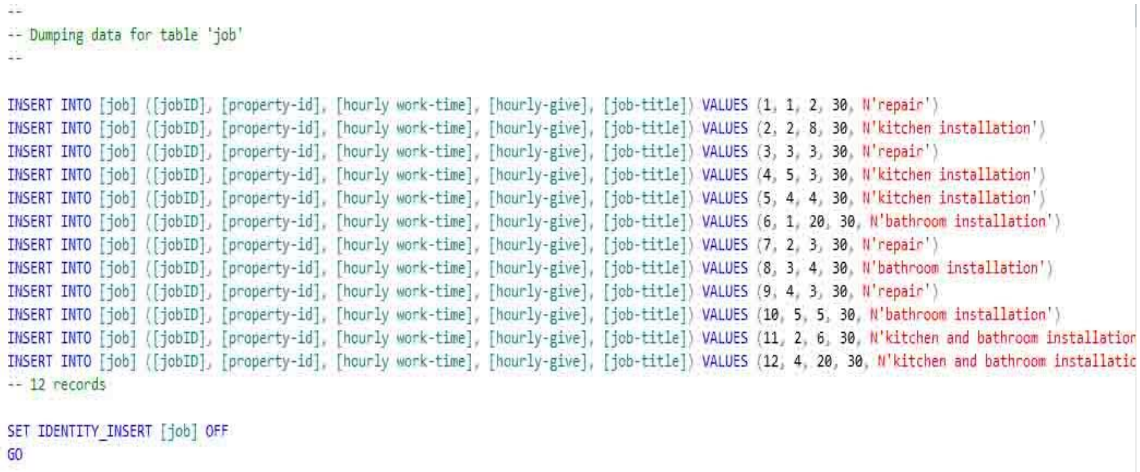
(2.1.3) Creating job table

SQL command “job” table



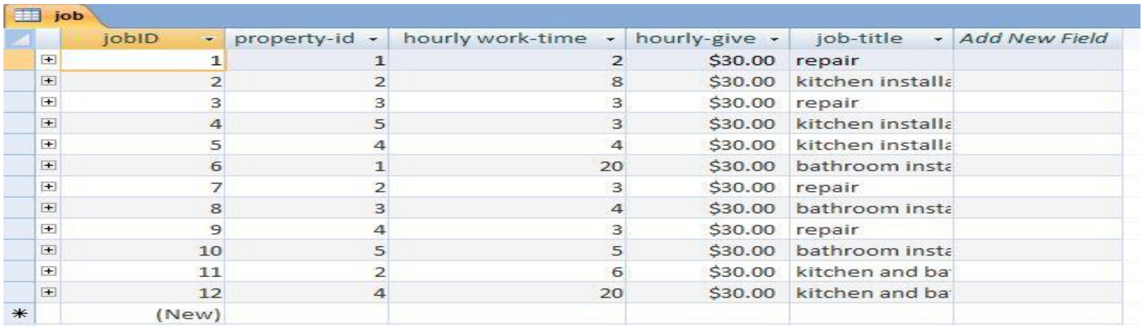
```
CREATE INDEX [fitting-id] ON [fitting] ([fitting-title])
GO

--
-- Table structure for table 'job'
--

IF object_id(N'job', 'U') IS NOT NULL DROP TABLE [job]

CREATE TABLE [job] (
   [jobID] INT NOT NULL IDENTITY,
   [property-id] INT,
   [hourly work-time] INT,
   [hourly-give] MONEY,
   [job-title] NVARCHAR(255),
   PRIMARY KEY ([jobID])
)

SET IDENTITY_INSERT [job] ON
GO
```

Value for job table



```
--
-- Dumping data for table 'job'
--

INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (1, 1, 2, 30, N'repair')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (2, 2, 8, 30, N'kitchen installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (3, 3, 3, 30, N'repair')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (4, 5, 3, 30, N'kitchen installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (5, 4, 4, 30, N'kitchen installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (6, 1, 20, 30, N'bathroom installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (7, 2, 3, 30, N'repair')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (8, 3, 4, 30, N'bathroom installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (9, 4, 3, 30, N'repair')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (10, 5, 5, 30, N'bathroom installation')
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (11, 2, 6, 30, N'kitchen and bathroom installation
INSERT INTO [job] ([jobID], [property-id], [hourly work-time], [hourly-give], [job-title]) VALUES (12, 4, 20, 30, N'kitchen and bathroom installatic
-- 12 records

SET IDENTITY_INSERT [job] OFF
GO
```

Output Result:



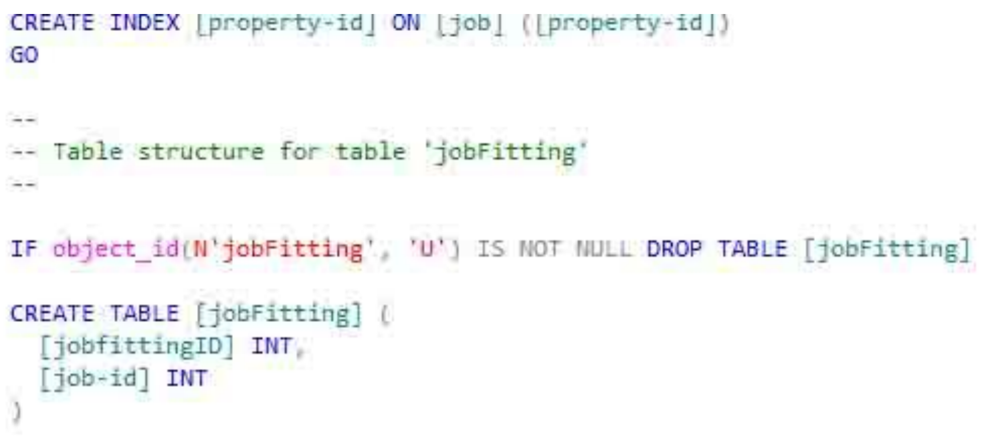
job

	jobID	property-id	hourly work-time	hourly-give	job-title	Add New Field
⊞	1	1	2	$30.00	repair	
⊞	2	2	8	$30.00	kitchen installa	
⊞	3	3	3	$30.00	repair	
⊞	4	5	3	$30.00	kitchen installa	
⊞	5	4	4	$30.00	kitchen installa	
⊞	6	1	20	$30.00	bathroom insta	
⊞	7	2	3	$30.00	repair	
⊞	8	3	4	$30.00	bathroom insta	
⊞	9	4	3	$30.00	repair	
⊞	10	5	5	$30.00	bathroom insta	
⊞	11	2	6	$30.00	kitchen and ba	
⊞	12	4	20	$30.00	kitchen and ba	
*	(New)					

Figure no: 2.3: Display of job table

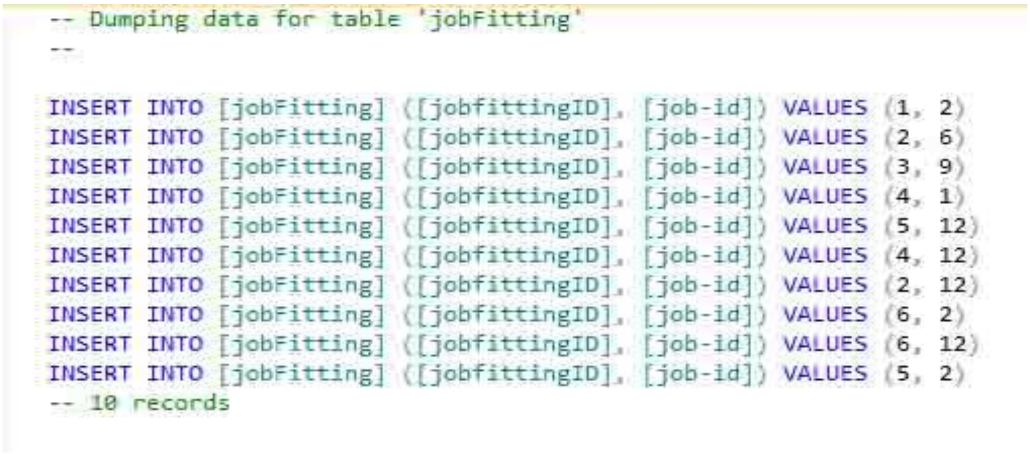
(2.1.4) Creating jobFitting table

SQL command for “jobFitting” table



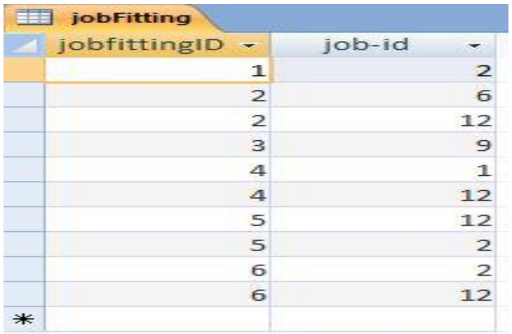
```
CREATE INDEX [property-id] ON [job] ([property-id])
GO

--
-- Table structure for table 'jobFitting'
--

IF object_id(N'jobFitting', 'U') IS NOT NULL DROP TABLE [jobFitting]

CREATE TABLE [jobFitting] (
  [jobfittingID] INT,
  [job-id] INT
)
```

Value for jobFitting table



```
-- Dumping data for table 'jobFitting'
--

INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (1, 2)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (2, 6)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (3, 9)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (4, 1)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (5, 12)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (4, 12)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (2, 12)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (6, 2)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (6, 12)
INSERT INTO [jobFitting] ([jobfittingID], [job-id]) VALUES (5, 2)
-- 10 records
```

Output Result:



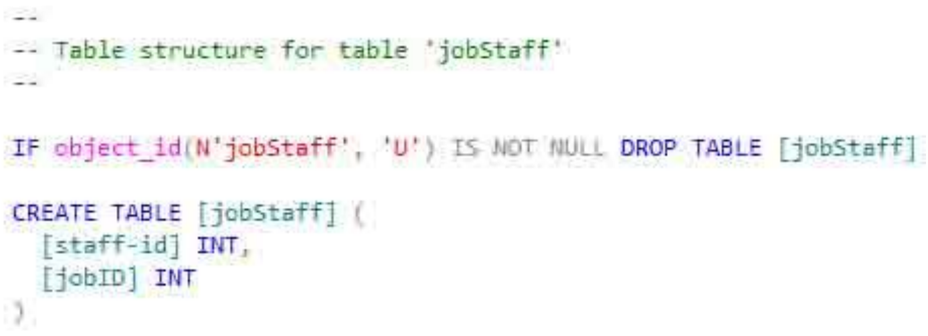
jobfittingID	job-id
1	2
2	6
2	12
3	9
4	1
4	12
5	12
5	2
6	2
6	12
*	

Figure no: 2.4: Display of jobFitting table

(Anon., n.d.)

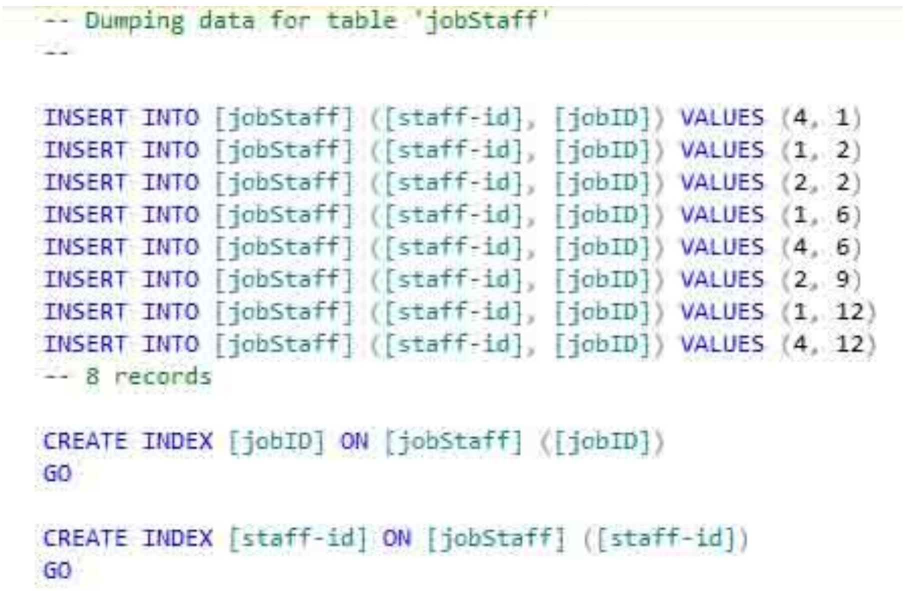
(2.1.5) Creating jobStaff table

SQL command for “jobStaff” table



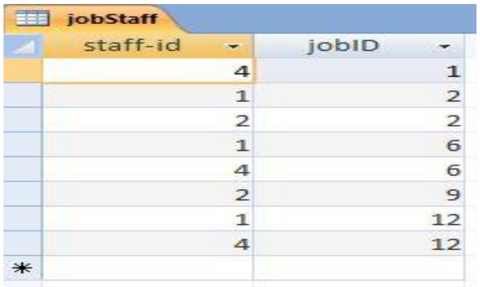
```
--
-- Table structure for table 'jobStaff'
--

IF object_id(N'jobStaff', 'U') IS NOT NULL DROP TABLE [jobStaff]

CREATE TABLE [jobStaff] (
  [staff-id] INT,
  [jobID] INT
);
```

Value for jobStaff table



```
-- Dumping data for table 'jobStaff'
--

INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (4, 1)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (1, 2)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (2, 2)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (1, 6)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (4, 6)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (2, 9)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (1, 12)
INSERT INTO [jobStaff] ([staff-id], [jobID]) VALUES (4, 12)
-- 8 records

CREATE INDEX [jobID] ON [jobStaff] ([jobID])
GO

CREATE INDEX [staff-id] ON [jobStaff] ([staff-id])
GO
```

Output Result:



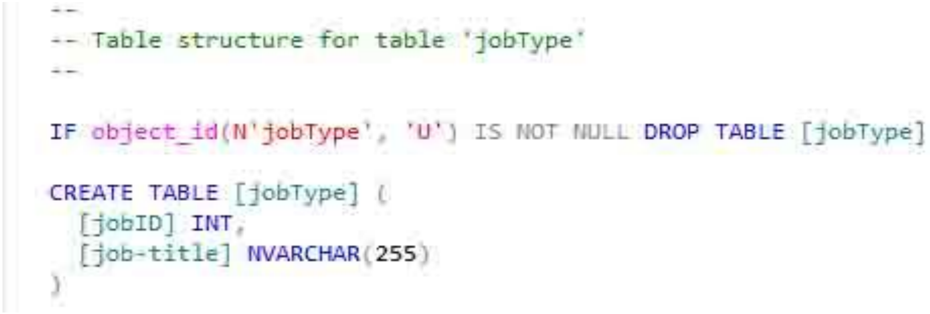
jobStaff

staff-id	jobID
4	1
1	2
2	2
1	6
4	6
2	9
1	12
4	12
*	

Figure no: 2.5: Display of jobStaff table

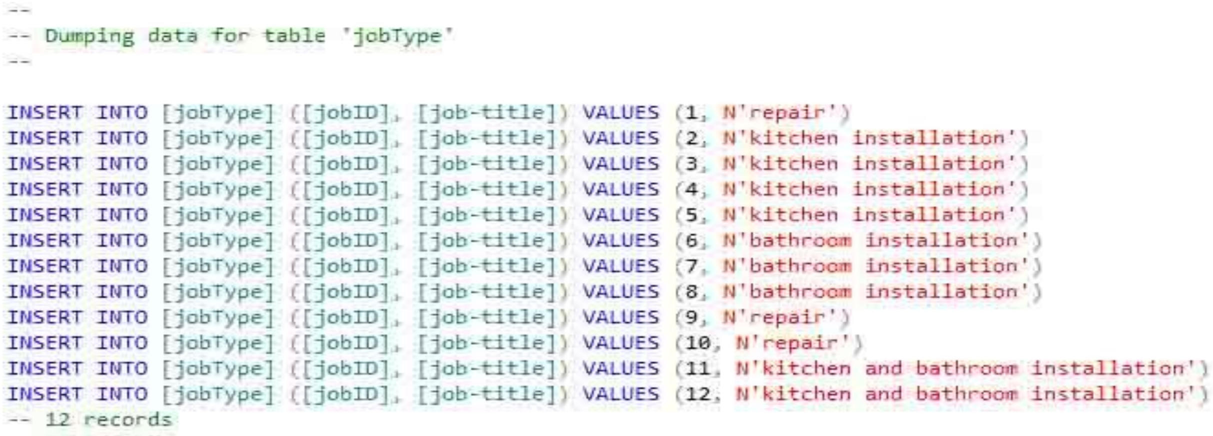
(2.1.6) Creating jobType table

SQL command for “jobType” table



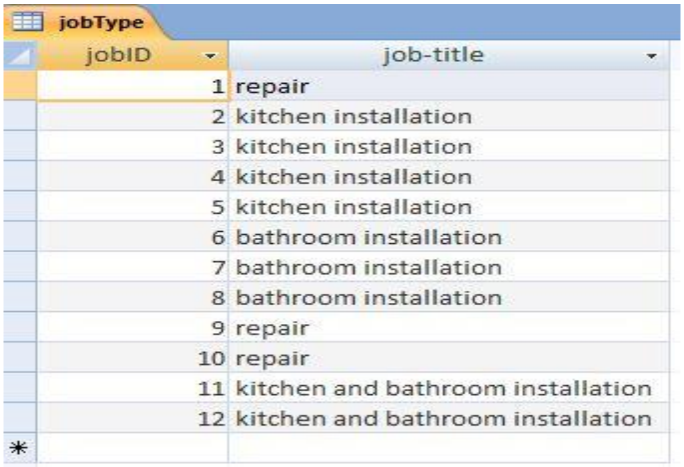
```
--
-- Table structure for table 'jobType'
--

IF object_id(N'jobType', 'U') IS NOT NULL DROP TABLE [jobType]

CREATE TABLE [jobType] (
  [jobID] INT,
  [job-title] NVARCHAR(255)
)
```

Value for jobType table



```
--
-- Dumping data for table 'jobType'
--

INSERT INTO [jobType] ([jobID], [job-title]) VALUES (1, N'repair')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (2, N'kitchen installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (3, N'kitchen installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (4, N'kitchen installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (5, N'kitchen installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (6, N'bathroom installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (7, N'bathroom installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (8, N'bathroom installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (9, N'repair')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (10, N'repair')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (11, N'kitchen and bathroom installation')
INSERT INTO [jobType] ([jobID], [job-title]) VALUES (12, N'kitchen and bathroom installation')
-- 12 records
```

Output Result:



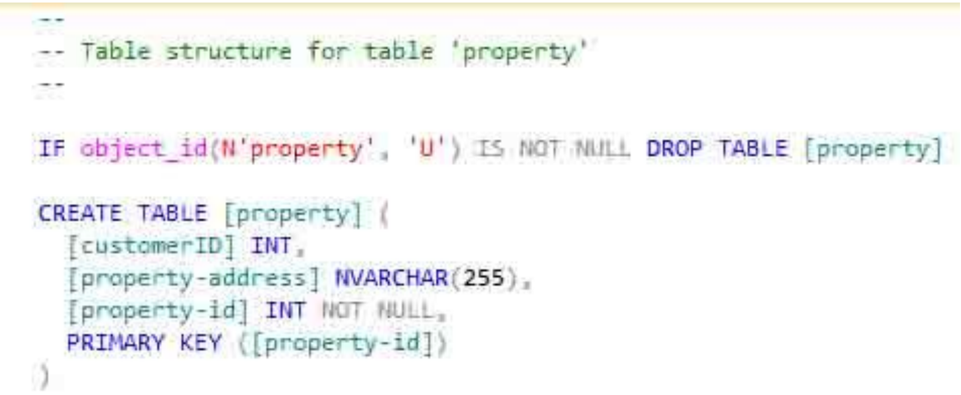
jobType

jobID	job-title
1	repair
2	kitchen installation
3	kitchen installation
4	kitchen installation
5	kitchen installation
6	bathroom installation
7	bathroom installation
8	bathroom installation
9	repair
10	repair
11	kitchen and bathroom installation
12	kitchen and bathroom installation
*	

Figure no: 2.6: Display of jobType table

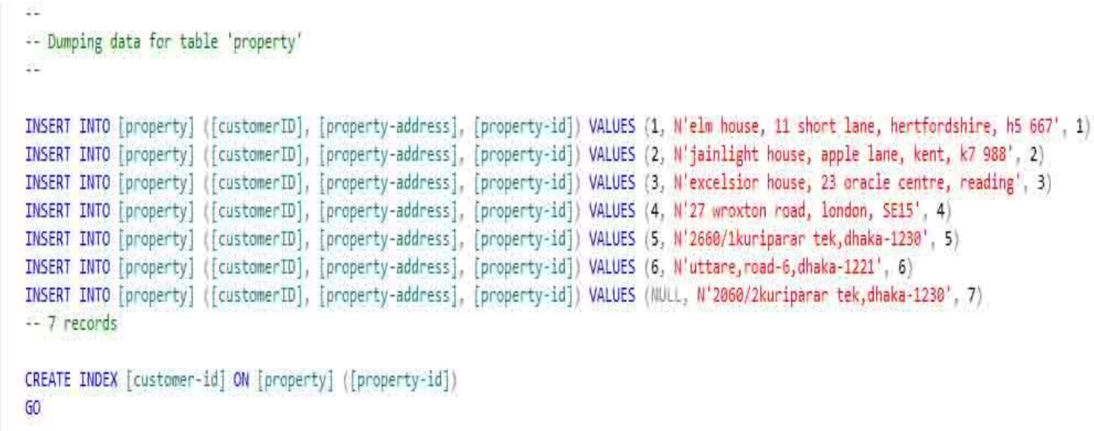
(2.1.7) Creating property table

SQL command for "property" table



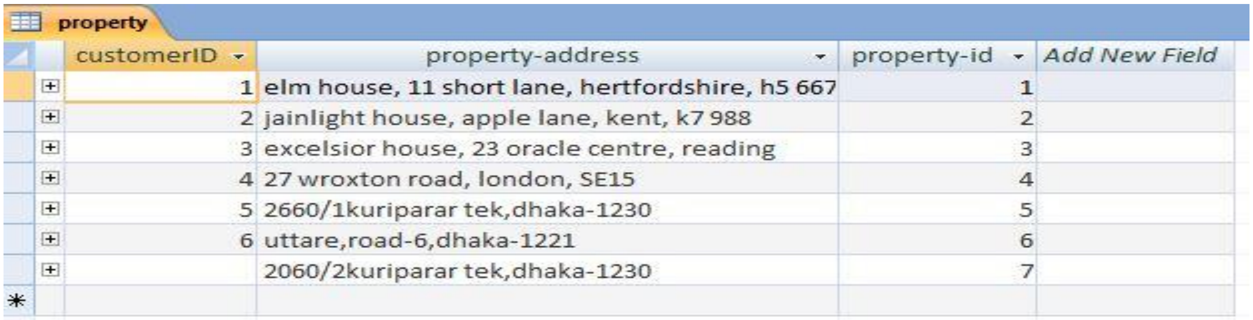
```
--
-- Table structure for table 'property'
--

IF object_id(N'property', 'U') IS NOT NULL DROP TABLE [property]

CREATE TABLE [property] (
  [customerID] INT,
  [property-address] NVARCHAR(255),
  [property-id] INT NOT NULL,
  PRIMARY KEY ([property-id])
)
```

Value for property table



```
--
-- Dumping data for table 'property'
--

INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (1, N'elm house, 11 short lane, hertfordshire, h5 667', 1)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (2, N'jainlight house, apple lane, kent, k7 988', 2)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (3, N'excelsior house, 23 oracle centre, reading', 3)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (4, N'27 wroxton road, london, SE15', 4)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (5, N'2660/1kuriparar tek,dhaka-1230', 5)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (6, N'uttare,road-6,dhaka-1221', 6)
INSERT INTO [property] ([customerID], [property-address], [property-id]) VALUES (NULL, N'2060/2kuriparar tek,dhaka-1230', 7)
-- 7 records

CREATE INDEX [customer-id] ON [property] ([property-id])
GO
```

Output Result:

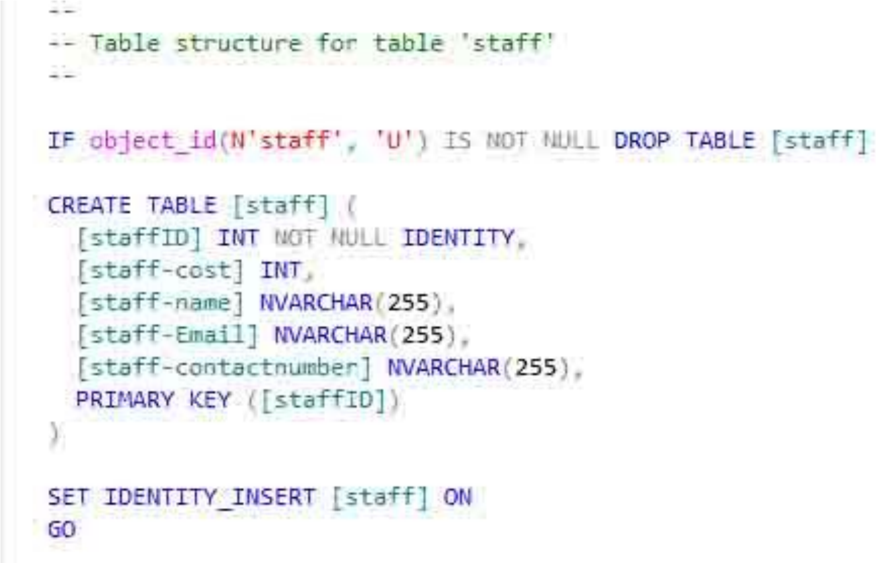


	customerID	property-address	property-id	Add New Field
⊞	1	elm house, 11 short lane, hertfordshire, h5 667	1	
⊞	2	jainlight house, apple lane, kent, k7 988	2	
⊞	3	excelsior house, 23 oracle centre, reading	3	
⊞	4	27 wroxton road, london, SE15	4	
⊞	5	2660/1kuriparar tek,dhaka-1230	5	
⊞	6	uttare,road-6,dhaka-1221	6	
⊞		2060/2kuriparar tek,dhaka-1230	7	
*				

Figure no: 2.7: Display of property table

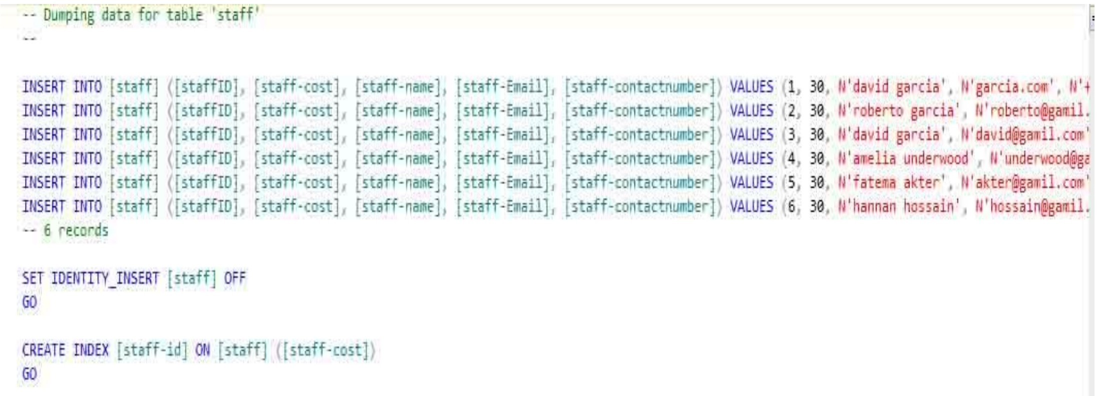
(2.1.8) Creating staff table

SQL command for “staff” table



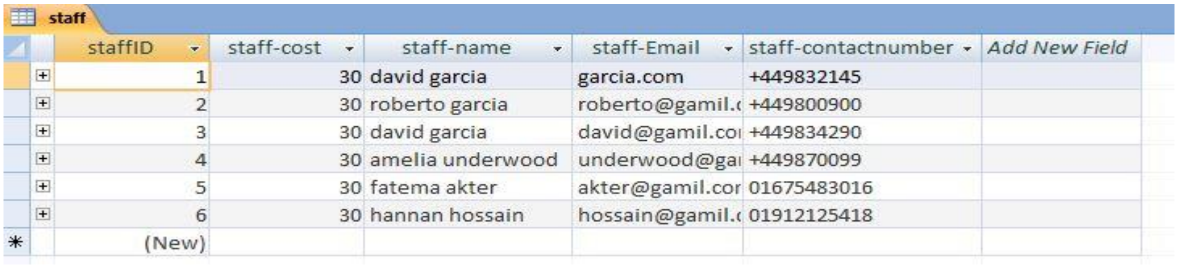
```
--
-- Table structure for table 'staff'
--

IF object_id(N'staff', 'U') IS NOT NULL DROP TABLE [staff]

CREATE TABLE [staff] (
  [staffID] INT NOT NULL IDENTITY,
  [staff-cost] INT,
  [staff-name] NVARCHAR(255),
  [staff-Email] NVARCHAR(255),
  [staff-contactnumber] NVARCHAR(255),
  PRIMARY KEY ([staffID])
)

SET IDENTITY_INSERT [staff] ON
GO
```

Value for staff table



```
-- Dumping data for table 'staff'
--

INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (1, 30, N'david garcia', N'garcia.com', N'+
INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (2, 30, N'roberto garcia', N'roberto@gamil.
INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (3, 30, N'david garcia', N'david@gamil.com'
INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (4, 30, N'amelia underwood', N'underwood@ga
INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (5, 30, N'fatema akter', N'akter@gamil.com'
INSERT INTO [staff] ([staffID], [staff-cost], [staff-name], [staff-Email], [staff-contactnumber]) VALUES (6, 30, N'hannan hossain', N'hossain@gamil.
-- 6 records

SET IDENTITY_INSERT [staff] OFF
GO

CREATE INDEX [staff-id] ON [staff] ([staff-cost])
GO
```

Output Result:



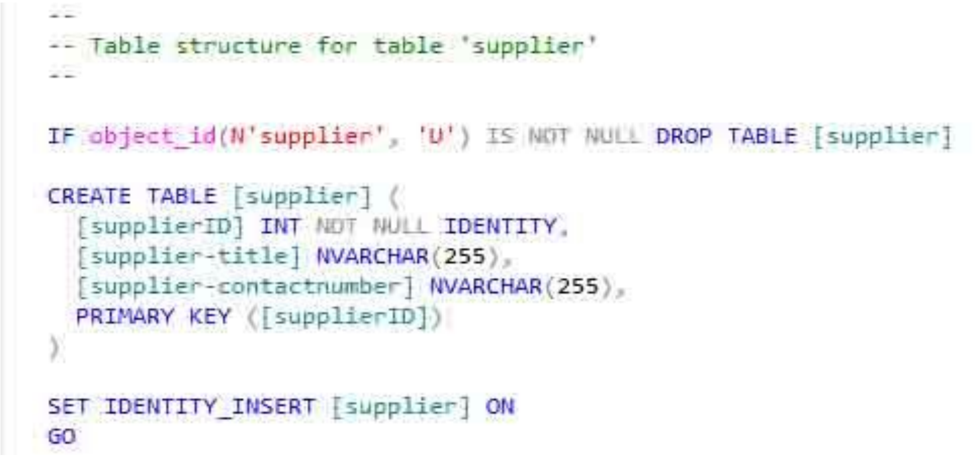
	staffID	staff-cost	staff-name	staff-Email	staff-contactnumber	Add New Field
⊞	1	30	david garcia	garcia.com	+449832145	
⊞	2	30	roberto garcia	roberto@gamil.c	+449800900	
⊞	3	30	david garcia	david@gamil.coı	+449834290	
⊞	4	30	amelia underwood	underwood@gaı	+449870099	
⊞	5	30	fatema akter	akter@gamil.coı	01675483016	
⊞	6	30	hannan hossain	hossain@gamil.c	01912125418	
*	(New)					

Figure no: 2.8: Display of staff table

(Anon., n.d.)

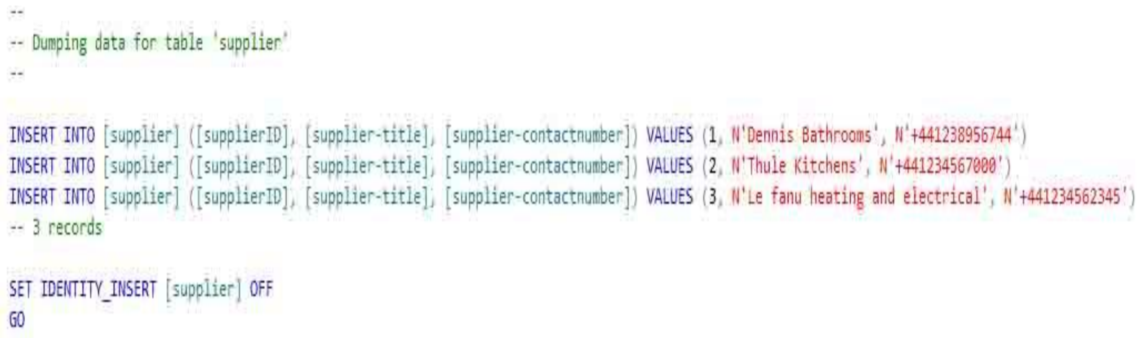
(2.1.9) Creating supplier table

SQL command for “supplier” table



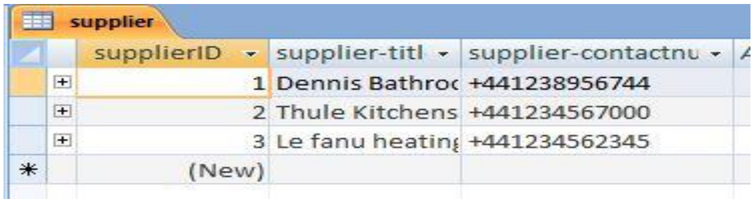
```
--
-- Table structure for table 'supplier'
--

IF object_id(N'supplier', 'U') IS NOT NULL DROP TABLE [supplier]

CREATE TABLE [supplier] (
  [supplierID] INT NOT NULL IDENTITY,
  [supplier-title] NVARCHAR(255),
  [supplier-contactnumber] NVARCHAR(255),
  PRIMARY KEY ([supplierID])
)

SET IDENTITY_INSERT [supplier] ON
GO
```

Value for supplier table



```
--
-- Dumping data for table 'supplier'
--

INSERT INTO [supplier] ([supplierID], [supplier-title], [supplier-contactnumber]) VALUES (1, N'Dennis Bathrooms', N'+441238956744')
INSERT INTO [supplier] ([supplierID], [supplier-title], [supplier-contactnumber]) VALUES (2, N'Thule Kitchens', N'+441234567000')
INSERT INTO [supplier] ([supplierID], [supplier-title], [supplier-contactnumber]) VALUES (3, N'Le fanu heating and electrical', N'+441234562345')
-- 3 records

SET IDENTITY_INSERT [supplier] OFF
GO
```

Output Result:

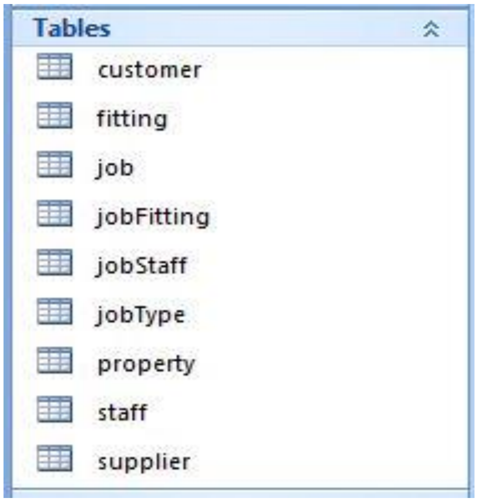


supplier

	supplierID	supplier-titl	supplier-contactnu	A
⊞	1	Dennis Bathroo	+441238956744	
⊞	2	Thule Kitchens	+441234567000	
⊞	3	Le fanu heating	+441234562345	
*	(New)			

Figure no: 2.9: Display of supplier table

Screenshot for database table:



Tables

- customer
- fitting
- job
- jobFitting
- jobStaff
- jobType
- property
- staff
- supplier

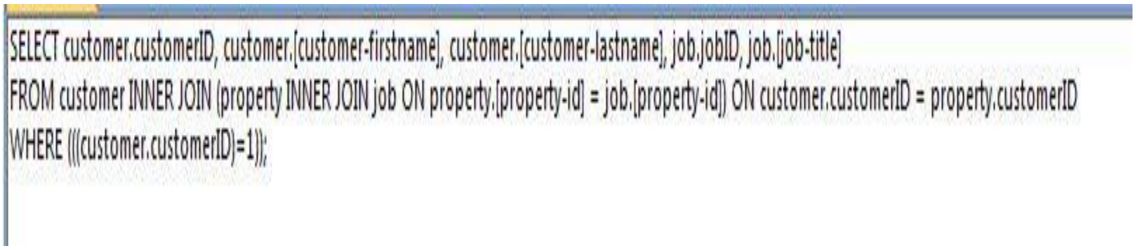
Figure no: 2.10: Display of all tables

(Anon., n.d.)

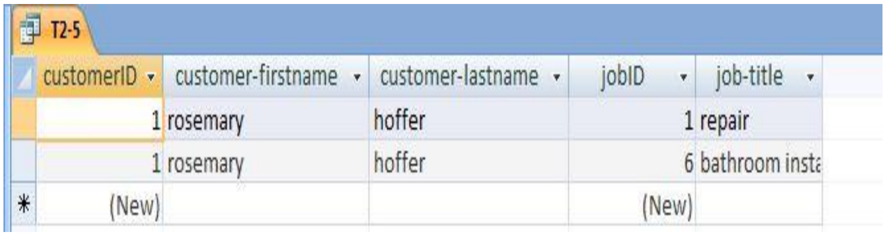
(2.2) Queries

(2.2.5) Query for selects all the jobs for a particular customer:

SQL code for selects all the jobs for a particular customer



```
SELECT customer.customerID, customer.[customer-firstname], customer.[customer-lastname], job.jobID, job.[job-title]
FROM customer INNER JOIN (property INNER JOIN job ON property.[property-id] = job.[property-id]) ON customer.customerID = property.customerID
WHERE (((customer.customerID)=1));
```



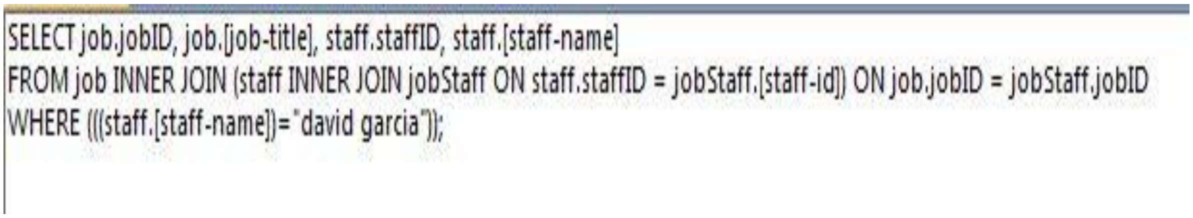
T2-5

	customerID	customer-firstname	customer-lastname	jobID	job-title
	1	rosemary	hoffer	1	repair
	1	rosemary	hoffer	6	bathroom insta
*	(New)			(New)	

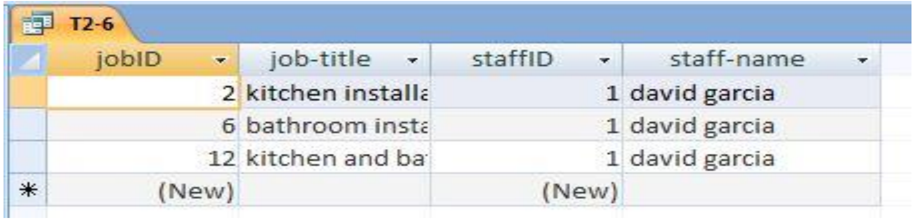
Figure no: 2.11: Display of queries jobs for a particular customer

(2.2.6) Query for selects the jobs that David Garcia has worked on:

SQL code for selects the jobs that David Garcia has worked on:



```
SELECT job.jobID, job.[job-title], staff.staffID, staff.[staff-name]
FROM job INNER JOIN (staff INNER JOIN jobStaff ON staff.staffID = jobStaff.[staff-id]) ON job.jobID = jobStaff.jobID
WHERE (((staff.[staff-name])="david garcia"));
```



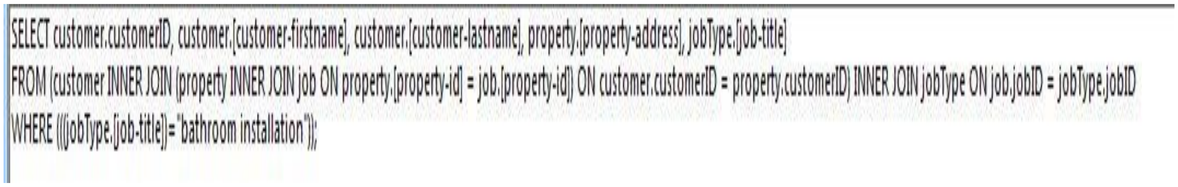
T2-6

jobID	job-title	staffID	staff-name
2	kitchen installa	1	david garcia
6	bathroom insta	1	david garcia
12	kitchen and ba	1	david garcia
(New)		(New)	

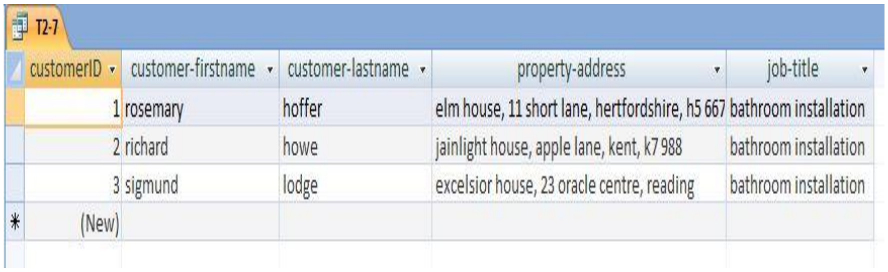
Figure no: 2.12: Display of queries jobs that David Garcia has worked on

(2.2.7) Query for selects the entire owner and addresses for bathroom installation jobs:

SQL code for selects the entire owner and addresses for bathroom installation jobs:



```
SELECT customer.customerID, customer.[customer-firstname], customer.[customer-lastname], property.[property-address], jobType.[job-title]
FROM (customer INNER JOIN (property INNER JOIN job ON property.[property-id] = job.[property-id]) ON customer.customerID = property.customerID) INNER JOIN jobType ON job.jobID = jobType.jobID
WHERE (((jobType.[job-title])="bathroom installation"));
```



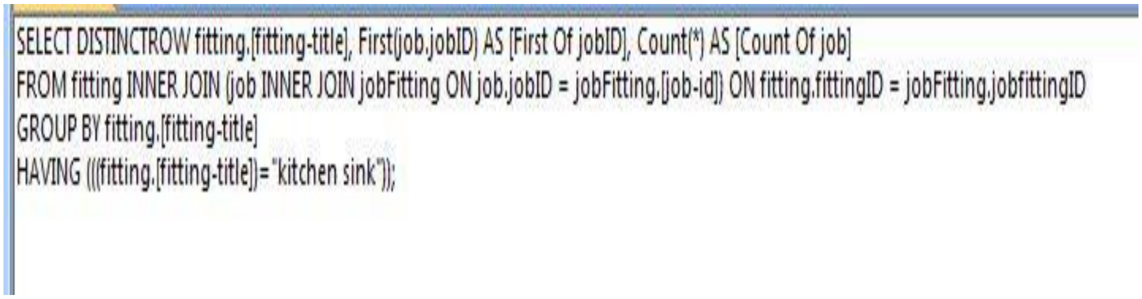
T2-7

customerID	customer-firstname	customer-lastname	property-address	job-title
1	rosemary	hoffer	elm house, 11 short lane, hertfordshire, h5 667	bathroom installation
2	richard	howe	jainlight house, apple lane, kent, k7 988	bathroom installation
3	sigmund	lodge	excelsior house, 23 oracle centre, reading	bathroom installation
(New)				

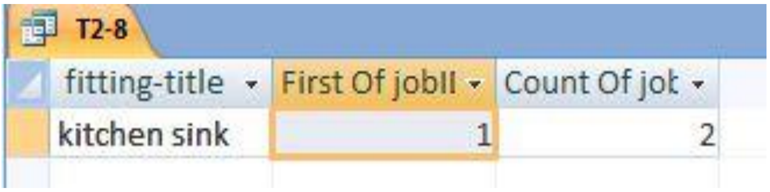
Figure no: 2.13: Display of queries owner and addresses for bathroom installation jobs

(2.2.8) Query for counts the numbers of jobs that need a kitchen sink:

SQL code for counts the numbers of jobs that need a kitchen sink:



```
SELECT DISTINCTROW fitting.[fitting-title], First(job.jobID) AS [First Of jobID], Count(*) AS [Count Of job]
FROM fitting INNER JOIN (job INNER JOIN jobFitting ON job.jobID = jobFitting.[job-id]) ON fitting.fittingID = jobFitting.jobfittingID
GROUP BY fitting.[fitting-title]
HAVING (((fitting.[fitting-title])="kitchen sink"));
```

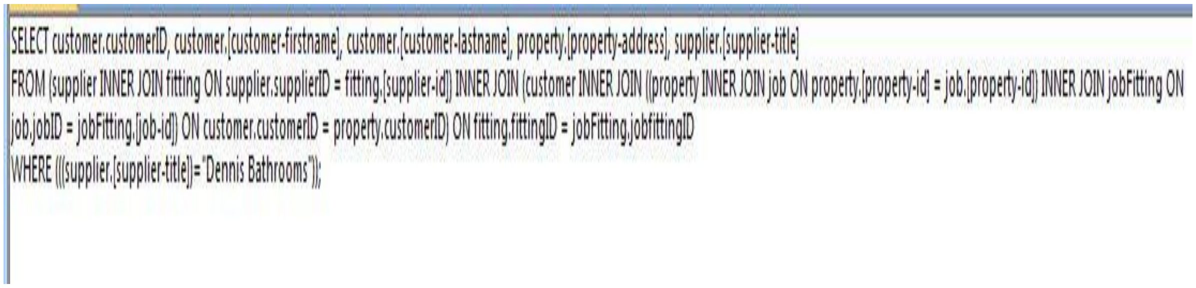


fitting-title	First Of jobI	Count Of jok
kitchen sink	1	2

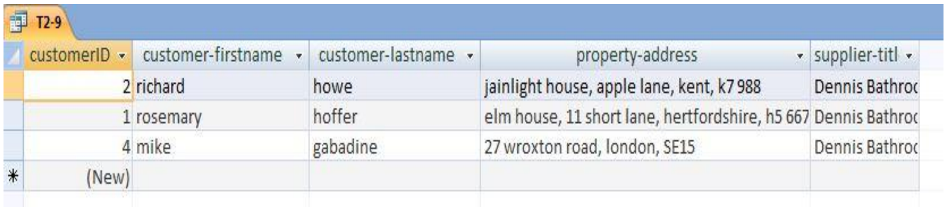
Figure no: 2.14: Display of queries number of jobs that need a kitchen sink

(2.2.9) Query for selects the customer names and addresses for all the jobs that require fittings from Dennis Bathrooms:

SQL code for selects the customer names and addresses for all the jobs:



```
SELECT customer.customerID, customer.[customer-firstname], customer.[customer-lastname], property.[property-address], supplier.[supplier-title]
FROM (supplier INNER JOIN fitting ON supplier.supplierID = fitting.[supplier-id]) INNER JOIN (customer INNER JOIN ((property INNER JOIN job ON property.[property-id] = job.[property-id]) INNER JOIN jobFitting ON
job.jobID = jobFitting.[job-id]) ON customer.customerID = property.customerID) ON fitting.fittingID = jobFitting.jobfittingID
WHERE (((supplier.[supplier-title])="Dennis Bathrooms"));
```

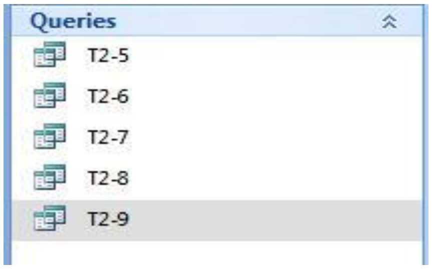


T2-9

customerID	customer-firstname	customer-lastname	property-address	supplier-titl
2	richard	howe	jainlight house, apple lane, kent, k7 988	Dennis Bathroo
1	rosemary	hoffer	elm house, 11 short lane, hertfordshire, h5 667	Dennis Bathroo
4	mike	gabadine	27 wroxton road, london, SE15	Dennis Bathroo
(New)				

Figure no: 2.15: Display of queries customer names and addresses

Queries table:

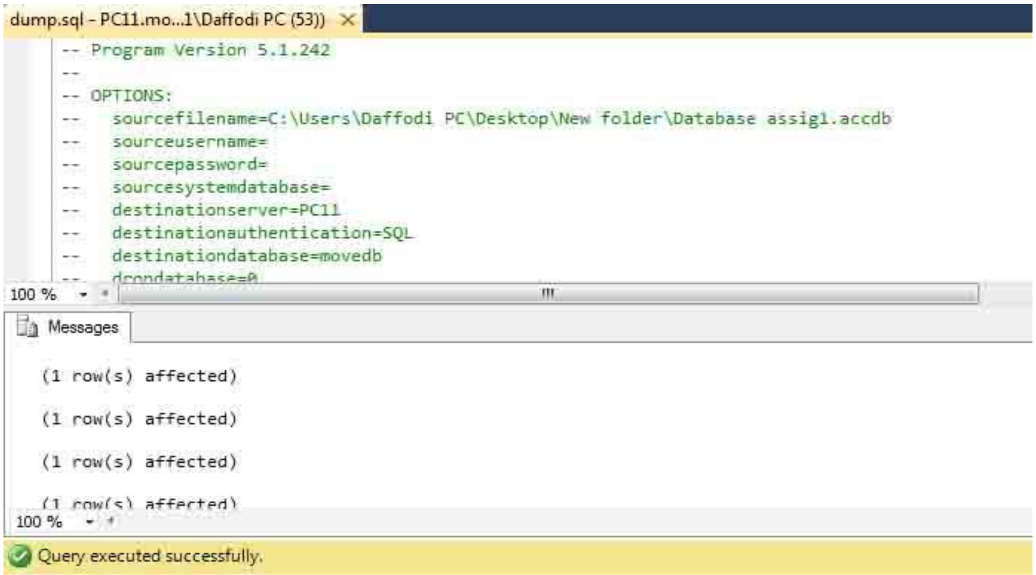


Queries

- T2-5
- T2-6
- T2-7
- T2-8
- T2-9

Figure no: 2.16: Display of queries table

Display for query executed successfully



dump.sql - PC11.mo...1\Daffodi PC (53))

```
-- Program Version 5.1.242
--
-- OPTIONS:
--   sourcefilename=C:\Users\Daffodi PC\Desktop\New folder\Database assig1.accdb
--   sourceusername=
--   sourcepassword=
--   sourcesystemdatabase=
--   destinationserver=PC11
--   destinationauthentication=SQL
--   destinationdatabase=movedb
--   dropdatabase=0
```

100 %

Messages

```
(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)
```

100 %

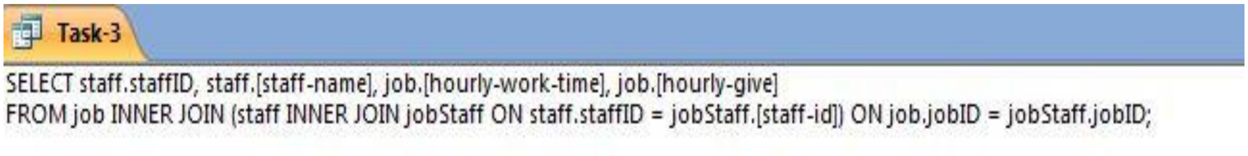
Query executed successfully.

Figure no: 2.17: Display of query table executed successfully

Task-3

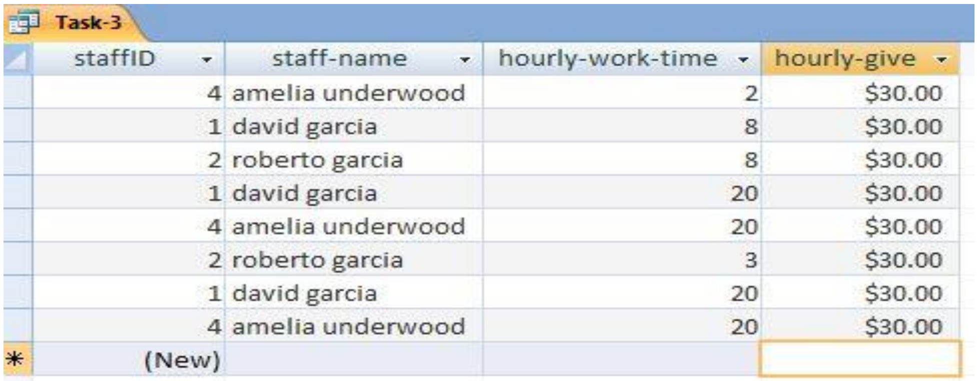
Costing queries:

SQL code for costing queries



```
SELECT staff.staffID, staff.[staff-name], job.[hourly-work-time], job.[hourly-give]
FROM job INNER JOIN (staff INNER JOIN jobStaff ON staff.staffID = jobStaff.[staff-id]) ON job.jobID = jobStaff.jobID;
```

Output Result:



staffID	staff-name	hourly-work-time	hourly-give
4	amelia underwood	2	$30.00
1	david garcia	8	$30.00
2	roberto garcia	8	$30.00
1	david garcia	20	$30.00
4	amelia underwood	20	$30.00
2	roberto garcia	3	$30.00
1	david garcia	20	$30.00
4	amelia underwood	20	$30.00
(New)			

Figure no: 3.1: Display of query costing

(Anon., n.d.)

Task -4

Assessment:

Assignment Requirement:

- Microsoft Access 2007
- Data table
- Data queries
- Data Dictionary
- Relationship
- Relationship model
- SQL

My attempt:

Firstly open Microsoft office access. Then click on new blank database. Then I named the file. After write file name, then click on create. A new table opened, now right clicks on table and selected on design view after show save as table name, write table name and click on ok. Then show table for field name and data type. After write field name and data type name. After finish again right click on table and selected datasheet view. Every table created this process. After created relationship for each other table. Then click on database tools and click on relationships after show every table. Selected table and click on add. After show every table. Now created relationship each other table. Click on one table and relation with another table that table is primary key and relationship with foreign key. So, relationship of one to many. There are many relationships with each other table. Like one to many, many to many. After created data and queries, from this file click on queries.

Result:

Task-1

- I have provided the documentation for organization of entity.
- I have completed the entity relationship diagram for Task-1.
- I have finished a data dictionary for all tables and values

Task-2

- Converted all the tables from Access into SQL.
- I have given all the screen shots for all the tables and queries in my assignment..

Task-3

- Query for costing selecting all the tables by a particular customer.
- Query for selecting the customers that a particular member of staff is assigned to via a maintenance contract.

- Queries select all the customers' details and installation information for customers to have a particular type of fitting in their set up (Anon., n.d.)

Conclusion:

In my assignment I have described the process with screenshot which I have followed for initial design of data base .I have completed data and queries and run successfully.

Bibliography

Anon., n.d. *http://www.info.com/convert%20access%20to%20sql?cb=21&cmp=4293*. [Online] Available at: http://www.info.com [Accessed 24 jun 2015].

Anon., n.d. *http://www.sqlcourse.com/intro.html*. [Online] Available at: http://www.sqlcourse.com [Accessed 23 jun 2015].

Anon., n.d. *http://www.thesitewizard.com/faqs/what-is-mysql-database.shtml*. [Online] Available at: http://www.thesitewizard.com [Accessed 21 jun 2015].

Anon., n.d. *https://msdn.microsoft.com/en-us/library/ms176061.aspx*. [Online] Available at: https://msdn.microsoft.com [Accessed 22 jun 2015].

Anon., n.d. *https://msdn.microsoft.com/en-us/library/ms178028.aspx*. [Online] Available at: https://msdn.microsoft.com [Accessed 22 jun 2015].

Anon., n.d. *https://www.youtube.com/watch?v=HCQpi1BC9Cg*. [Online] Available at: https://www.youtube.com [Accessed 25 jun 2015].

http://www.thesitewizard.com/faqs/what-is-mysql-database.shtml, n.d. [Online] Available at: http://www.thesitewizard.com [Accessed 21 jun 2015].