



A JOB-SHOP ACCOUNTING SYSTEM

Akhil Sanjay Potdar | 11346860 | akhilpotdar@ou.edu



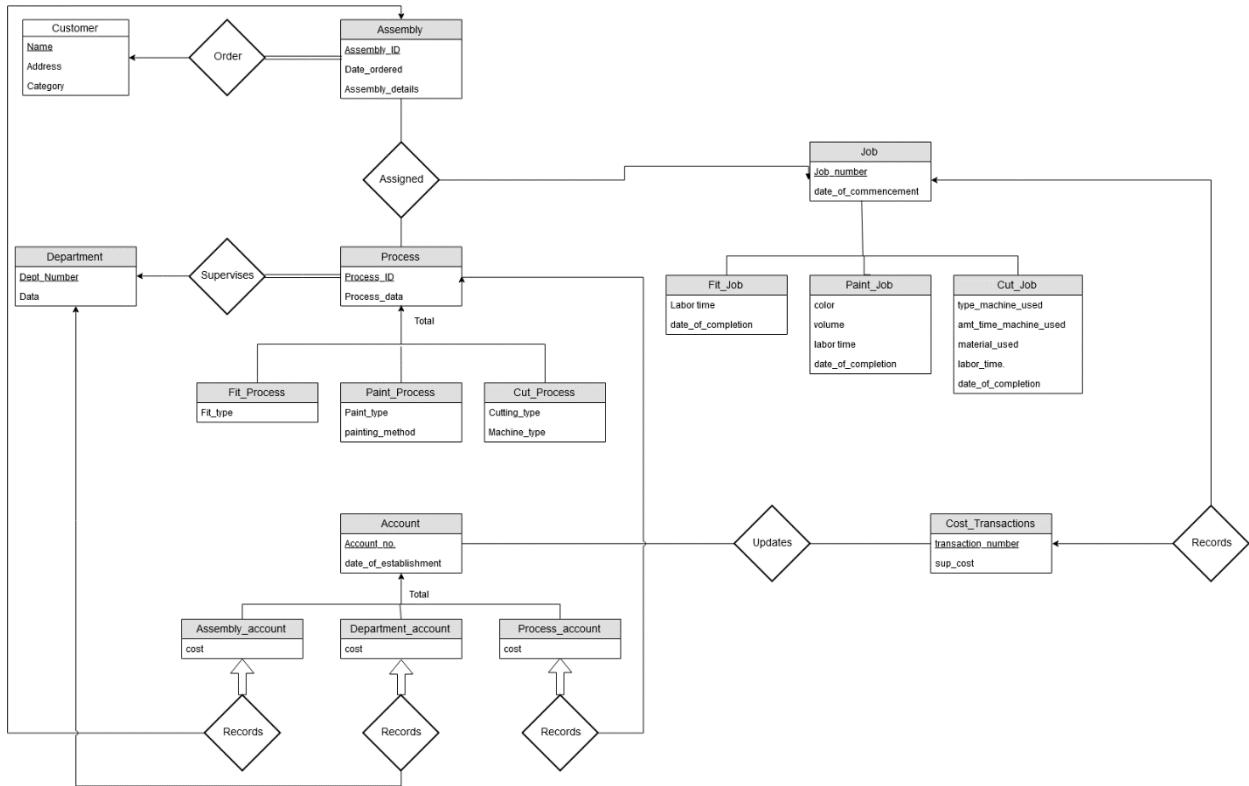
NOVEMBER 19, 2019
DATABASE MANAGEMENT SYSTEM – CS/DSA 4513-001
Dr.Lee Grunwall

Contents

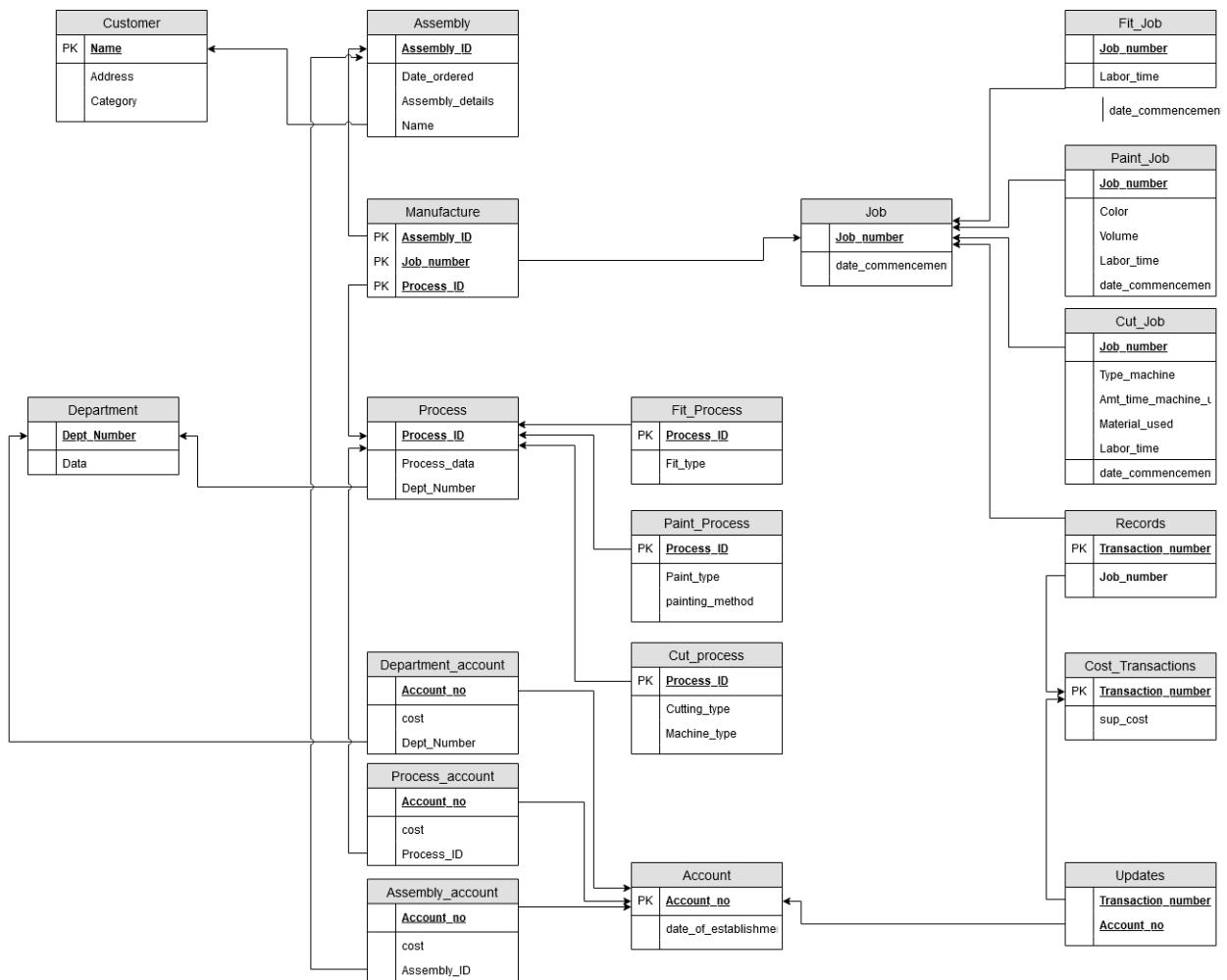
| | |
|---|-----|
| Task 1..... | 3 |
| ER Diagram..... | 3 |
| Relational Database Schema..... | 4 |
| Task 2. Data Dictionary | 5 |
| Task 3..... | 7 |
| Discussion of storage structures for tables..... | 7 |
| Discussion of storage structures for tables (Azure SQL Database) | 7 |
| Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database | 9 |
| Task 5. The Java source program and screenshots showing its successful compilation..... | 30 |
| Task 6. Java program Execution | 76 |
| Screenshots showing the testing of query 1 | 76 |
| Screenshots showing the testing of query 2 | 77 |
| Screenshots showing the testing of query 3 | 78 |
| Screenshots showing the testing of query 4 | 79 |
| Screenshots showing the testing of query 5 | 80 |
| Screenshots showing the testing of query 6 | 81 |
| Screenshots showing the testing of query 7 | 82 |
| Screenshots showing the testing of query 8 | 83 |
| Screenshots showing the testing of query 9 | 85 |
| Screenshots showing the testing of query 10 | 85 |
| Screenshots showing the testing of query 11 | 86 |
| Screenshots showing the testing of query 12 | 87 |
| Screenshots showing the testing of query 13 | 87 |
| Screenshots showing the testing of query 14 | 88 |
| Screenshots showing the testing of query 15 | 88 |
| Screenshots showing the testing of the import and export options | 90 |
| Screenshots showing the testing of three types of errors | 92 |
| Screenshots showing the testing of the quit option..... | 93 |
| Task 7. Web database application and its execution..... | 94 |
| Web database application source program and screenshots showing Its successful compilation..... | 94 |
| Screenshots showing the testing of the Web database application | 103 |

Task 1.

ER Diagram



Relational Database Schema



Task 2. Data Dictionary

| Tables | Attributes | Data Type | Size (bytes) | Constraints |
|--------------------|-----------------------|-----------|--------------|-----------------------|
| Customer | Name | varchar | 27 | PK |
| | Address | varchar | 27 | |
| | Category | int | 4 | Not Null;Between 0-10 |
| Department | Dept_Number | int | 4 | PK |
| | Data | varchar | 27 | |
| Assembly | Assembly_ID | varchar | 27 | PK |
| | Date_ordered | Date | 3 | |
| | Assembly_details | varchar | 27 | |
| | Name | varchar | 27 | FK |
| Process | Process_ID | varchar | 27 | PK |
| | Process_data | varchar | 27 | |
| | Dept_Number | int | 4 | FK |
| Account | Account_no | int | 4 | PK |
| | date_of_establishment | Date | 3 | |
| Assembly_account | Account_no | int | 4 | PK,FK |
| | cost | int | 4 | |
| | Assembly_ID | varchar | 27 | FK |
| Department_account | Account_no | int | 4 | PK,FK |
| | cost | int | 4 | |
| | Dept_Number | int | 4 | FK |
| Process_account | Account_no | int | 4 | PK,FK |
| | cost | int | 4 | |
| | Process_ID | varchar | 27 | FK |
| Job | Job_number | int | 4 | PK |
| | date_commencement | Date | 3 | |
| Assigned | Job_number | int | 4 | PK,FK |
| | Assembly_ID | varchar | 27 | PK,FK |
| | Process_ID | varchar | 27 | PK,FK |
| Fit_Job | Job_number | int | 4 | PK,FK |
| | Labor_time | Time | 5 | Not Null |
| | date_completeion | | | |
| Paint_Job | Job_number | int | 4 | PK,FK |
| | Color | nvarchar | 102 | Not Null |
| | Volume | numeric | 5 to 17 | Not Null |
| | Labor_time | Time | 5 | Not Null |
| | date_completeion | Date | 3 | |
| Cut_Job | Job_number | int | 4 | PK,FK |
| | Type_machine_used | nvarchar | 102 | Not Null |
| | Amt_time_machine_used | Time | 5 | Not Null |
| | Material_used | nvarchar | 102 | Not Null |
| | Labor_time | Time | 5 | Not Null |
| | date_completeion | Date | 3 | |
| Fit_Process | Process_ID | varchar | 27 | PK,FK |
| | Fit_type | varchar | 27 | |
| Paint_Process | Process_ID | varchar | 27 | PK,FK |
| | Paint_type | varchar | 27 | |
| | Painting_method | varchar | 27 | |
| Cut_process | Process_ID | varchar | 27 | PK,FK |
| | Cutting_type | varchar | 27 | |
| | Machine_type | varchar | 27 | |
| Records | Transaction_number | int | 4 | Not Null, PK,FK |
| | Job_number | int | 4 | Not Null, FK |
| Cost_Transactions | Transaction_number | int | 4 | Not Null, PK |
| | sup_cost | int | 4 | Not Null, PK |
| Updates | Transaction_number | int | 4 | Not Null, PK,FK |
| | Account_no | int | 4 | Not Null, PK,FK |

6

Task 3.

Discussion of storage structures for tables

Discussion of storage structures for tables (Azure SQL Database)

In the above table we consider only one type of indexing option, which is based on the frequency of occurrence of the queries. As such, each table will be having only one index, based on the utility. No changes to the above.

| Tables | Query# | Type | Search Key | Query Frequency | Selected File Organization | | Justification |
|--------------------|--------|---------------|-----------------|-----------------|----------------------------|-----------------|---|
| | | | | | Type | Search Key | |
| Customer | 1 | Insertion | | 30/day | | | |
| | 13 | Range Search | Category | 100/day | | Category | Good for range search and accomodates insertion |
| Department | 2 | Insertion | | infrequent | | | |
| Assembly | 3 | Insertion | | 40/day | | | |
| | 9 | Random Search | Assembly_ID | 200/day | Extendable Hashing | Assembly_ID | Good for random search and accomodates insertion |
| Process | 4 | Insertion | | infrequent | | | |
| | 10 | Random Search | Department No. | 20/day | | | |
| | 11 | Random Search | Process-ID | 100/day | Extendable Hashing | Process-ID | Good for random search and accomodates insertion |
| | 12 | Random Search | Department No. | 20/day | | | |
| Account | 5 | Insertion | | 10/day | | | |
| | 8 | Random Search | Account No | 50/day | Extendable Hashing | Account No | Good for random search and random search outweighs insertion |
| Assembly_account | 5 | Insertion | | 10/day | | | |
| | 8 | Random Search | Assembly-ID | 50/day | Extendable Hashing | Assembly-ID | Good for random search and accomodates insertion |
| Department_account | 5 | Insertion | | 10/day | | | |
| | 8 | Random Search | Department-ID | 50/day | Extendable Hashing | Department-ID | Good for random search and accomodates insertion |
| Process_account | 5 | Insertion | | 10/day | | | |
| | 8 | Random Search | Process-ID | 50/day | Extendable Hashing | Process-ID | Good for random search and accomodates insertion |
| Job | 6 | Insertion | | 50/day | | | |
| | 7 | Insertion | | 50/day | | | |
| Assigned | 10 | Random Search | Process-ID | 20/day | | | |
| | 8 | Random Search | Job Number | 50/day | | | |
| | 11 | Random Search | Assembly-ID | 100/day | Extendable Hashing | Assembly-ID | Good for random search on Assembly ID is greater than others |
| | 12 | Random Search | Process-ID | 20/day | | | |
| Fit_Job | 6 | Insertion | | 50/day | Extendable Hashing | Job Number | Good for random search and accomodates insertion |
| | 10 | Random Search | Job Number | 20/day | | | |
| | 12 | Random Search | Job Number | 20/day | | | |
| Paint_Job | 6 | Insertion | | 50/day | Extendable Hashing | Job Number | Good for random search and accomodates insertion |
| | 10 | Random Search | Job Number | 20/day | | | |
| Cut_Job | 6 | Insertion | | 50/day | | | |
| | 10 | Random Search | Job Number | 20/day | B+ Tree | Job Number | Good for range search and accomodates insertion |
| | 13 | Range Search | Job Number | 1/month | | | |
| Cost_Transactions | 8 | Insertion | | 50/day | Heap | | Since no foreign key and only insertions |
| Updates | 8 | Insertion | | 50/day | | | |
| Records | 8 | Insertion | | 50/day | | | Good for random search and random and is helpful in insertion too |
| | 8 | Random Search | Transaction no. | 50/day | Extendable Hashing | Transaction no. | |
| Fit_Process | 4 | Insertion | | infrequent | | | |
| Paint_Process | 4 | Insertion | | infrequent | | | |
| Cut_process | 4 | Insertion | | infrequent | | | |

Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database

```
DROP TABLE IF EXISTS Updates, Records, Cost_Transactions
DROP TABLE IF EXISTS Assembly_account, Process_account, Department_account
DROP TABLE IF EXISTS Account
DROP PROCEDURE IF EXISTS assembly_account_entry,department_account_entry,process_account_entry
DROP TABLE IF EXISTS Fit_Job, Paint_Job, Cut_Job, Fit_Process, Paint_Process, Cut_Process
DROP TABLE IF EXISTS Assigned, Job, Assembly, Process
DROP TABLE IF EXISTS Customer,Department
DROP PROCEDURE IF EXISTS job_entry,assembly_entry,process_entry;
DROP PROCEDURE IF EXISTS customer_entry,department_entry;
DROP PROCEDURE IF EXISTS account_entry, labor_time_retrieve;
DROP PROCEDURE IF EXISTS fit_job_completion, paint_job_completion, cut_job_completion, retrieve_process
drop PROCEDURE if exists cost_transaction_entry, record_entry, updates_entry, cost_assembly_retrieve
drop PROCEDURE if exists fit_job_retrieve,paint_job_retrieve, cut_job_retrieve,customer_retrieve
drop PROCEDURE if exists cut_job_deletion,color_change, cost_transaction_record_entry
```

```
CREATE TABLE Customer(
Name varchar(25) PRIMARY KEY,
Address varchar(25),
Category int not null check (Category between 0 and 10),
);
create index c on Customer(Category)

CREATE TABLE Department (
Dept_Number int PRIMARY KEY,
Data varchar(25));
```

```
CREATE TABLE Assembly (
Assembly_ID varchar(25) PRIMARY KEY,
Date_ordered DATE,
Assembly_details varchar(25),
Name varchar(25),
FOREIGN KEY(Name) REFERENCES Customer,
);
create index d on Assembly(Assembly_ID)
```

```
CREATE TABLE Process (
Process_ID varchar(25) PRIMARY KEY,
Process_data varchar(25),
Dept_Number int,
FOREIGN KEY(Dept_Number) REFERENCES Department,
);
create index e on process(process_ID)
```

```

CREATE TABLE Account (
Account_no int PRIMARY KEY,
Date_of_establishment DATE,
);

Create index f on Account(Account_no)

CREATE TABLE Assembly_account (
Account_no int PRIMARY KEY,
cost int,
Assembly_ID VARCHAR(25),
FOREIGN KEY(Account_no) REFERENCES Account,
FOREIGN KEY(Assembly_ID) REFERENCES Assembly,
);
create index g on Assembly_account(Assembly_ID)

CREATE TABLE Department_account (
Account_no int PRIMARY KEY,
cost int,
Dept_Number int,
FOREIGN KEY(Account_no) REFERENCES Account,
FOREIGN KEY(Dept_Number) REFERENCES Department,
);
create index h on Department_account(Dept_Number)

CREATE TABLE Process_account (
Account_no int PRIMARY KEY,
cost int,
Process_ID VARCHAR(25),
FOREIGN KEY(Account_no) REFERENCES Account,
FOREIGN KEY(Process_ID) REFERENCES Process,
);
create index i on Process_account(Process_ID)

CREATE TABLE Job (
Job_number int PRIMARY KEY,
date_commencement DATE);

CREATE TABLE Assigned (
Job_number int,
Assembly_ID varchar(25),
Process_ID varchar(25),
FOREIGN KEY(Job_number) REFERENCES Job,
FOREIGN KEY(Assembly_ID) REFERENCES Assembly,
FOREIGN KEY(Process_ID) REFERENCES Process,
PRIMARY KEY(Job_number,Assembly_ID,Process_ID),
);
create index j on Assigned(Assembly_ID)

CREATE TABLE [Fit_Job]
(
[Job_number] INT Primary Key, -- Primary Key column
[Labor_time] TIME NOT NULL,

```

```

[date_completion] DATE,
FOREIGN KEY(Job_number) REFERENCES Job,
index name(Job_Number)
);

CREATE TABLE [Paint_Job]
(
    [Job_number] INT Primary Key, -- Primary Key column
[Color] NVARCHAR(50) NOT NULL,
[Volume] NUMERIC NOT NULL,
[Labor_time] TIME NOT NULL,
[date_completion] DATE,
FOREIGN KEY(Job_number) REFERENCES Job,
index name(Job_Number)
);

CREATE TABLE [Cut_Job]
(
    [Job_number] INT Primary Key, -- Primary Key column
[Type_machine_used] NVARCHAR(50) NOT NULL,
[Amt_time_machine_used] TIME NOT NULL,
[Material_used] NVARCHAR(50) NOT NULL,
[Labor_time] TIME NOT NULL,
[date_completion] DATE,
FOREIGN KEY(Job_number) REFERENCES Job
);
Create Index a on Cut_Job(Job_Number)

CREATE TABLE Cost_Transactions
(
    [Transaction_number] INT NOT NULL PRIMARY KEY, -- Primary Key column
[sup_cost] int NOT NULL
);

CREATE TABLE [Records]
(
    [Transaction_number] INT NOT NULL Primary Key,
[Job_number] INT NOT NULL,
FOREIGN KEY(Transaction_number) REFERENCES Cost_Transactions,
FOREIGN KEY(Job_number) REFERENCES Job,
);

CREATE INDEX b on Records(Transaction_number)

CREATE TABLE Updates
(
    [Transaction_number] INT NOT NULL,
[Account_no] int NOT NULL,
FOREIGN KEY(Transaction_number) REFERENCES Cost_Transactions,
FOREIGN KEY(Account_no) REFERENCES Account,
Primary Key(Transaction_number,Account_no)
);

```

```

CREATE TABLE [Fit_Process]
(
    [Process_ID] varchar(25) Primary Key, -- Primary Key column
    [Fit_type] varchar(25) NOT NULL,
    FOREIGN KEY(Process_ID) REFERENCES Process
);

CREATE TABLE [Paint_Process]
(
    [Process_ID] varchar(25) Primary Key, -- Primary Key column
    [Type] varchar(25) NOT NULL,
    [Method] NUMERIC NOT NULL,
    FOREIGN KEY(Process_ID) REFERENCES Process
);

CREATE TABLE [Cut_Process]
(
    [Process_ID] varchar(25) Primary Key, -- Primary Key column
    [Cutting_Type] varchar(25) NOT NULL,
    [Machine_type] varchar(25) NOT NULL,
    FOREIGN KEY(Process_ID) REFERENCES Process
);

```

```

GO
CREATE PROCEDURE customer_entry
    @name varchar(25),
    @add varchar(25),
    @cat int
AS
BEGIN
    INSERT INTO Customer(Name, Address, Category)
    VALUES (@name, @add, @cat)
END

```

```

-- Query 2 --
-- Creating a Department table
GO
CREATE PROCEDURE department_entry
    @num int,
    @data varchar(25)
AS
BEGIN
    INSERT INTO Department(Dept_Number,[Data])
    VALUES (@num, @data)
END

-- Query 3 --
-- Creating a Assembly table
GO
CREATE PROCEDURE assembly_entry

```

```

@id varchar(25),
@date DATE,
@details varchar(25),
@name varchar(25)
AS
BEGIN
INSERT INTO Assembly(Assembly_ID,[Date_ordered],Assembly_details,Name)
VALUES (@id, @date, @details, @name)
END

-- Query 4 --
-- Creating a Process table

GO
CREATE PROCEDURE process_entry
@id varchar(25),
@details varchar(25),
@num int
AS
BEGIN
INSERT INTO Process(Process_ID,Process_data,Dept_Number)
VALUES (@id,@details, @num)
END

-- Query 5 --

GO
CREATE PROCEDURE account_entry
@num int,
@date DATE
AS
BEGIN
INSERT INTO Account (Account_no, Date_of_establishment)
VALUES (@num,@date)
END

GO
CREATE PROCEDURE assembly_account_entry
@num int,
@cost int,
@a_ID VARCHAR(25)
AS
BEGIN
INSERT INTO Assembly_account (Account_no,cost,Assembly_ID)
VALUES (@num,@cost,@a_ID)
END

GO
CREATE PROCEDURE department_account_entry
@num int,
@cost int,

```

```

@d_num int
AS
BEGIN
INSERT INTO Department_account (Account_no, cost,Dept_Number)
VALUES (@num,@cost,@d_num)
END

GO
CREATE PROCEDURE process_account_entry
@num int,
@cost int,
@p_ID VARCHAR(25)
AS
BEGIN
INSERT INTO Process_account (Account_no, cost,Process_ID)
VALUES (@num,@cost,@p_ID)
END

--Query 6 --
-- Creating a Job table

-- Creating a Assigned table
GO
CREATE PROCEDURE job_entry
@num int,
@date DATE,
@a_id VARCHAR(25),
@p_id VARCHAR(25)
AS
BEGIN
INSERT INTO Job(Job_number,date_commencement)
VALUES (@num,@date)
INSERT INTO Assigned(Job_number,Assembly_ID,Process_ID)
VALUES (@num,@a_id,@p_id)
END

--Query 7--
-- Create the table in the specified schema

GO
CREATE PROCEDURE fit_job_completion
@num int,
@l_time TIME,
@date DATE
AS
BEGIN
INSERT INTO Fit_Job(Job_number,Labor_time,date_completion)
VALUES (@num,@l_time,@date)
END

GO
CREATE PROCEDURE paint_job_completion
@num int,

```

```

@col VARCHAR(25),
@vol NUMERIC,
@l_time TIME,
@date DATE
AS
BEGIN
INSERT INTO Paint_Job(Job_number,Color,Volume,Labor_time,date_completion)
VALUES (@num,@col,@vol,@l_time,@date)
END

GO
CREATE PROCEDURE cut_job_completion
@num int,
@typ VARCHAR(25),
@a_time TIME,
@mat VARCHAR(25),
@l_time TIME,
@date DATE
AS
BEGIN
INSERT INTO Cut_Job(Job_number,Type_machine_used,Amt_time_machine_used,
Material_used,Labor_time,date_completion)
VALUES (@num,@typ,@a_time,@mat,@l_time,@date)
END
--Query 8--

GO
CREATE PROCEDURE cost_transaction_record_entry
@t_num int,
@s_cost int,
@j_num int
AS
BEGIN
INSERT INTO Cost_Transactions(Transaction_number,sup_cost)
VALUES (@t_num,@s_cost)
INSERT INTO Records(Transaction_number,Job_number)
VALUES (@t_num,@j_num)
END

GO
CREATE PROCEDURE updates_entry
@t_num int
AS
BEGIN
--Retrieving Process account number
DECLARE @p_acc int
SET @p_acc = (SELECT P.Account_no
from Process_account P
where Process_ID in
(SELECT Process_ID
From Assigned
Where Job_Number in

```

```

(select Job_number from Records
where Transaction_number = @t_num)))

DECLARE @a_acc int
SET @a_acc =(SELECT A.Account_no
from Assembly_account A
Where Assembly_ID in
(SELECT Assembly_ID
From Assigned
Where Job_Number in
(select Job_number from Records
where Transaction_number = @t_num)))

DECLARE @d_acc int
SET @d_acc = (SELECT Distinct D.Account_no
from Department_account D
Where Dept_Number IN
(SELECT Dept_Number
From Process
where Process_ID IN
(Select Process_ID
From Assigned
Where Job_Number in
(select Job_number from Records
where Transaction_number = @t_num)))))

INSERT INTO Updates(Transaction_number,Account_no)
VALUES (@t_num,@p_acc)
INSERT INTO Updates(Transaction_number,Account_no)
VALUES (@t_num,@a_acc)
INSERT INTO Updates(Transaction_number,Account_no)
VALUES (@t_num,@d_acc)

DECLARE @s_cost int
SET @s_cost = (SELECT sup_cost FROM Cost_Transactions where Transaction_number=@t_num)

--SELECT sup_cost FROM Cost_Transactions where Transaction_number=1

update Assembly_account
set cost = cost + @s_cost
where Account_no=@a_acc

update Department_account
set cost = cost + @s_cost
where Account_no=@d_acc

update Process_account
set cost = cost + @s_cost
where Account_no=@p_acc
END

```

-- Query 9

```

GO
CREATE PROCEDURE cost_assembly_retrieve
    @a_id VARCHAR(25)
AS
BEGIN
SELECT cost
from Assembly_account
where Assembly_ID = @a_id
END

--Query 10
drop PROCEDURE if exists labor_time_retrieve
GO
CREATE PROCEDURE labor_time_retrieve
    @end DATE,
    @dept int
AS
BEGIN
-- Searching for fit_job
DECLARE @fit int
SET @fit = 0
SET @fit = (SELECT SUM(DATEDIFF(MINUTE, '0:00:00', Labor_time))
from Fit_job where date_completion=@end and Job_number in(SELECT Job_number
from Assigned
WHERE Process_ID in (select Process_ID
from Process
where Dept_Number=@dept)))
PRINT @fit
--Seaching for paint_job
DECLARE @paint int
SET @paint = 0
SET @paint = (SELECT SUM(DATEDIFF(MINUTE, '0:00:00', Labor_time))
from Paint_job where date_completion='2019-02-01' and Job_number in(SELECT Job_number
from Assigned
WHERE Process_ID in (select Process_ID
from Process
where Dept_Number=201)))
PRINT @paint
--Select for cut_job
DECLARE @cut int
SET @cut = 0
SET @cut = (SELECT SUM(DATEDIFF(MINUTE, '0:00:00', Labor_time))
from Cut_job where date_completion='2019-02-01' and Job_number in(SELECT Job_number
from Assigned
WHERE Process_ID in (select Process_ID
from Process
where Dept_Number=201)))
PRINT @cut
DECLARE @tot int
SET @tot = @fit+@cut+@paint
--Select @tot

```

```
Print (@tot)
END
```

--Query 11

```
GO
CREATE PROCEDURE retrieve_process
    @a_id VARCHAR(25)
AS
BEGIN
    select P.Process_id, P.Dept_Number , J.date_commencement
    from Process P,Job J,Assigned AA
    where J.Job_number = AA.Job_number and AA.Process_ID = P.Process_ID and P.Process_id in
    (select A.Process_ID
    from Assigned A
    where A.Assembly_ID='a_id')
    order by J.date_commencement
END

select distinct P.Process_id, P.Dept_Number, J.date_commencement
from Process P,Job J
where P.Process_id in
(select distinct A.Job_number
from Assigned A
where A.Assembly_ID='A2')
order by J.date_commencement

select distinct J.Job_number,J.date_commencement
from Process P,Job J
where P.Process_ID in
(select distinct A.Process_ID
from Assigned A
where A.Assembly_ID='A2') AND
J.Job_number in
(select distinct A.Job_number
from Assigned A
where A.Assembly_ID='A2')
order by J.date_commencement;
```

--Query 12

```
/* Retrieve the jobs (together with their type information and assembly-id)
completed during a given date in a given department (20/day) */
```

```
GO
CREATE PROCEDURE fit_job_retrieve
    @end DATE,
    @dept int
AS
BEGIN
    SELECT distinct F.* ,A.Assembly_ID
```

```

from Fit_Job F, Assigned A
where F.date_completion=@end and
F.Job_number in
    (SELECT A.Job_number
     from Assigned A
     WHERE A.Process_ID in
        (select P.Process_ID
         from Process P
         where P.Dept_Number=@dept
        )
    ) AND
A.Assembly_ID in
    (SELECT A.Assembly_ID
     from Assigned A
     WHERE A.Process_ID in
        (select P.Process_ID
         from Process P
         where P.Dept_Number=@dept
        ) AND
A.Job_number = F.Job_number
    )
END

GO
CREATE PROCEDURE paint_job_retrieve
    @end DATE,
    @dept int
AS
BEGIN
SELECT distinct F.* ,A.Assembly_ID
from Paint_Job F, Assigned A
where F.date_completion=@end and
F.Job_number in
    (SELECT A.Job_number
     from Assigned A
     WHERE A.Process_ID in
        (select P.Process_ID
         from Process P
         where P.Dept_Number=@dept
        )
    ) AND
) AND
A.Assembly_ID in
    (SELECT A.Assembly_ID
     from Assigned A
     WHERE A.Process_ID in
        (select P.Process_ID
         from Process P
         where P.Dept_Number=@dept
        ) AND
A.Job_number = F.Job_number
    )
END

```

GO

```

CREATE PROCEDURE cut_job_retrieve
    @end DATE,
    @dept int
AS
BEGIN
SELECT distinct F.* ,A.Assembly_ID
from Cut_Job F, Assigned A
where F.date_completion=@end and
F.Job_number in
    (SELECT A.Job_number
    from Assigned A
    WHERE A.Process_ID in
        (select P.Process_ID
        from Process P
        where P.Dept_Number=@dept
        )
    ) AND
A.Assembly_ID in
    (SELECT A.Assembly_ID
    from Assigned A
    WHERE A.Process_ID in
        (select P.Process_ID
        from Process P
        where P.Dept_Number=@dept
        ) AND
A.Job_number = F.Job_number
    )
END

```

```

--Query 13
GO
CREATE PROCEDURE customer_retrieve
    @start int,
    @end int
AS
BEGIN
select *
from Customer
where Category >=@start and Category<=@end
order by Name
END

go
EXECUTE customer_retrieve @start=1, @end=5
select * from Customer
--Query 14
GO
CREATE PROCEDURE cut_job_deletion
    @start int,
    @end int
AS
BEGIN
delete from Cut_Job

```

```

where Job_number>=@start and Job_number<=@end
END

go
--EXECUTE cut_job_deletion @start=4, @end=7
select * from Cut_Job

--Query 15
GO
CREATE PROCEDURE color_change
@col NVARCHAR(50),
@num int
AS
BEGIN
update Paint_Job
set Color=@col
where Job_number=@num
END

GO
EXEC customer_entry @name = 'Naveen', @add = 'Norman', @cat = 1;
EXEC customer_entry @name = 'Taras', @add = 'Dallas', @cat = 3;
EXEC customer_entry @name = 'Ryan', @add = 'OKC', @cat = 5;
EXEC customer_entry @name = 'Jack', @add = 'Chicago', @cat = 8;
EXEC customer_entry @name = 'Bryan', @add = 'SFO', @cat = 6;
EXEC customer_entry @name = 'Akhil', @add = 'Norman', @cat = 8;

GO
EXEC department_entry @num = 101, @data = 'Soldering';
EXEC department_entry @num = 201, @data = 'Smelting';
EXEC department_entry @num = 301, @data = 'Forging';
EXEC department_entry @num = 401, @data = 'Artistic';
EXEC department_entry @num = 501, @data = 'Hydraulic';
EXEC department_entry @num = 601, @data = 'Repairs';

GO
EXEC assembly_entry @id = 'A1', @date='1-AUG-17', @details='Ordered a big assembly',
@name='Naveen'
EXEC assembly_entry @id = 'A2', @date='1-June-18', @details='To get car doors',
@name='Taras'
EXEC assembly_entry @id='A3', @date='08-01-18', @details='Smelting unit',
@name='Akhil'

GO
EXEC process_entry @id = 'P1', @details='Cutting car doors', @num=101
EXEC process_entry @id = 'P2', @details='Glazing car doors', @num=201
EXEC process_entry @id = 'P3', @details='Creating moulds for cars', @num=301

GO
EXEC account_entry @num=1001, @date='1-Aug-17'
EXEC account_entry @num=1002, @date='1-Oct-17'

```

```

EXEC account_entry @num=1003, @date='1-Nov-17'
EXEC account_entry @num=1004, @date='12-1-17'
EXEC account_entry @num=1005, @date='01-01-18'
EXEC account_entry @num=1006, @date='02-01-18'
EXEC account_entry @num=1007, @date='03-01-18'
EXEC account_entry @num=1008, @date='04-01-18'
EXEC account_entry @num=1009, @date='05-01-18'

GO
EXEC assembly_account_entry @num=1001, @cost=50, @a_id='A1'
EXEC assembly_account_entry @num=1005, @cost=35, @a_id='A2'
EXEC assembly_account_entry @num=1008, @cost=65, @a_id='A3'
EXEC department_account_entry @num=1002, @cost=10,@d_num=101
EXEC department_account_entry @num=1006, @cost=10,@d_num=201
EXEC department_account_entry @num=1007, @cost=10,@d_num=301
EXEC process_account_entry @num=1003, @cost=25, @p_ID='P1'
EXEC process_account_entry @num=1004, @cost=15, @p_ID='P2'
EXEC process_account_entry @num=1009, @cost=45, @p_ID='P3'

GO
EXEC job_entry @num=1,@date='1-Oct-2018',@a_id='A2',@p_id='P1'
EXEC job_entry @num=2,@date='1-Nov-2018',@a_id='A2',@p_id='P2'
EXEC job_entry @num=3,@date='1-Dec-2018',@a_id='A1',@p_id='P2'
EXEC job_entry @num=4,@date='01-01-2019',@a_id='A1',@p_id='P1'
EXEC job_entry @num=5,@date='02-01-2019',@a_id='A3',@p_id='P3'
EXEC job_entry @num=6,@date='03-01-2019',@a_id='A3',@p_id='P1'
EXEC job_entry @num=7,@date='03-01-2019',@a_id='A3',@p_id='P2'
EXEC job_entry @num=8,@date='04-01-2019',@a_id='A1',@p_id='P3'

GO
EXEC fit_job_completion @num=1,@l_time='1:30',@date='1-Jan-2019'
EXEC fit_job_completion @num=5,@l_time='2:30',@date='1-Feb-2019'
EXEC paint_job_completion @num=2,@col='Black',
    @vol=50,@l_time='2:30',@date='1-Feb-2019'
EXEC paint_job_completion @num=6,@col='Yellow',
    @vol=50,@l_time='4:00',@date='04-01-2019'
EXEC paint_job_completion @num=8,@col='Red',
    @vol=30,@l_time='2:00',@date='05-01-2019'
EXEC cut_job_completion @num=3,@typ='Cutter',
    @a_time='2:50',@mat='iron sheet',@l_time='2:30',@date='2-March-2019'
EXEC cut_job_completion @num=4,@typ='Cutting bumpers',@a_time="1:30",
    @mat='Coated steel',@l_time='3:00',@date='04-01-2019'
EXEC cut_job_completion @num=7,@typ='Cutting tyres',@a_time="2:30",
    @mat='Vulcanized Rubber',@l_time='2:50',@date='05-01-2019'

GO
EXEC cost_transaction_record_entry @t_num=1, @s_cost=10, @j_num=1
EXEC cost_transaction_record_entry @t_num=2, @s_cost=20, @j_num=2
EXEC cost_transaction_record_entry @t_num=3, @s_cost=15, @j_num=3

```

Go

```
EXEC retrieve_process @a_id='A2'

GO
EXECUTE updates_entry @t_num='1'
EXECUTE updates_entry @t_num='2'
EXECUTE updates_entry @t_num='3'

Go
EXECUTE cost_assembly_retrieve @a_id='A2'

GO
EXECUTE labor_time_retrieve @end='2019-02-01',@dept=201

GO
EXECUTE fit_job_retrieve @end='2019-01-01',@dept=101
EXECUTE paint_job_retrieve @end='2019-02-01',@dept=201
EXECUTE cut_job_retrieve @end='2019-03-02', @dept= 201

go
EXECUTE color_change @col='Green', @num=2
select * from Paint_Job
```

File Edit View Help • Table_Creation.sql - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

SQL_Query.sql - disconnected • SQLQuery_1 - disconnected • Main_SQLsql - potd00...td0000 • Table_Creation.sql - potd00...td0000 •

C: > Users > akhil > OneDrive - University of Oklahoma > Subjects > DBMS > Individual Project > Table_Creation.sql

Run Cancel ⚡ Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

17 CREATE TABLE Customer (
18     Name varchar(25) PRIMARY KEY,
19     Address varchar(25),
20     Category int not null check (Category between 0 and 10),
21 );
22 create index c on Customer(category)

23 CREATE TABLE Department (
24     Dept_Number int PRIMARY KEY,
25     Data varchar(25));
26

28 CREATE TABLE Assembly (
29     Assembly_ID varchar(25) PRIMARY KEY,
30     Date_ordered DATE,
31     Assembly_details varchar(25),
32     Name varchar(25),
33     FOREIGN KEY(Name) REFERENCES Customer,
34 );
35 create index d on Assembly(Assembly_ID)

37 CREATE TABLE Process (
38     Process_ID varchar(25) PRIMARY KEY,
39     Process_data varchar(25),
40     Dept_Number int,
41     FOREIGN KEY(Dept_Number) REFERENCES Department,
42 );
43 create index e on process(process_ID)

45 CREATE TABLE Account (
46     Account_no int PRIMARY KEY,
47     Date_of_establishment DATE,
48 );
49

50 Create index f on Account(Account_no)

```

Ln 5, Col 90 Spaces: 4 UTF-8 CRLF SQL Choose SQL Language potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG 9:18 PM IN 11/19/2019

File Edit View Help • Table_Creation.sql - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

SQL_Query.sql - disconnected • SQLQuery_1 - disconnected • Main_SQLsql - potd00...td0000 • Table_Creation.sql - potd00...td0000 •

C: > Users > akhil > OneDrive - University of Oklahoma > Subjects > DBMS > Individual Project > Table_Creation.sql

Run Cancel ⚡ Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

50 Create index f on Account(Account_no)

51 CREATE TABLE Assembly_account (
52     Account_no int PRIMARY KEY,
53     cost int,
54     Assembly_ID VARCHAR(25),
55     FOREIGN KEY(Account_no) REFERENCES Account,
56     FOREIGN KEY(Assembly_ID) REFERENCES Assembly,
57 );
58 create index g on Assembly_account(Assembly_ID)

61 CREATE TABLE Department_account (
62     Account_no int PRIMARY KEY,
63     cost int,
64     Dept_Number int,
65     FOREIGN KEY(Account_no) REFERENCES Account,
66     FOREIGN KEY(Dept_Number) REFERENCES Department,
67 );
68 create index h on Department_account(Dept_Number)

70 CREATE TABLE Process_Account (
71     Account_no int PRIMARY KEY,
72     cost int,
73     Process_ID VARCHAR(25),
74     FOREIGN KEY(Account_no) REFERENCES Account,
75     FOREIGN KEY(Process_ID) REFERENCES Process,
76 );
77 create index i on Process_Account(Process_ID)

79 CREATE TABLE Job (
80     Job_number int PRIMARY KEY,
81     date_commencement DATE);
82

83 CREATE TABLE Assigned (

```

Ln 5, Col 90 Spaces: 4 UTF-8 CRLF SQL Choose SQL Language potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG 9:18 PM IN 11/19/2019

The screenshot shows a Microsoft Data Studio interface with the following details:

- Title Bar:** File, Edit, View, Help, Table_Creation.sql - ptd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (ptd0000) - Azure Data Studio
- Toolbar:** SQL_Query.sql - disconnected, SQLQuery_1 - disconnected, Main_SQLsql - ptd0000...td0000, Table_Creation.sql - ptd0000...td0000.
- Search Bar:** C: > Users > akhil > OneDrive - University of Oklahoma > Subjects > DBMS > Individual Project > Table_Creation.sql
- Run Options:** Run, Cancel, Disconnect, Change Connection (cs-dsa-4513-sql-db), Explain, Enable SQLCMD.
- Code Area:** The code block contains 166 lines of T-SQL script for creating four tables: Assigned, Fit_Job, Paint_Job, and Cut_Job. The code includes primary keys, foreign keys, and indexes.
- Status Bar:** Ln 5, Col 90, Spaces: 4, UTF-8, CRLF, SQL, Choose SQL Language, ptd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db, ENG, 9:18 PM, IN, 11/19/2019.

```
83 CREATE TABLE Assigned (
84     Job_number int,
85     Assembly_ID varchar(25),
86     Process_ID varchar(25),
87     FOREIGN KEY(Job_number) REFERENCES Job,
88     FOREIGN KEY(Assembly_ID) REFERENCES Assembly,
89     FOREIGN KEY(Process_ID) REFERENCES Process,
90     PRIMARY KEY(Job_number,Assembly_ID,Process_ID),
91 );
92 create index j on Assigned(Assembly_ID)
93
94 CREATE TABLE [Fit_Job]
95 (
96     [Job number] INT Primary Key, -- Primary Key column
97     [Labor_time] TIME NOT NULL,
98     [date_completion] DATE,
99     FOREIGN KEY(Job_number) REFERENCES Job,
100    index name(Job_Number)
101 );
102
103 CREATE TABLE [Paint_Job]
104 (
105     [Job_number] INT Primary Key, -- Primary Key column
106     [color] NVARCHAR(50) NOT NULL,
107     [Volume] NUMERIC NOT NULL,
108     [Labor_time] TIME NOT NULL,
109     [date_completion] DATE,
110     FOREIGN KEY(Job_number) REFERENCES Job,
111     index name(Job_Number)
112 );
113
114 CREATE TABLE [Cut_Job]
115 (
116     [Job number] INT Primary Key, -- Primary Key column
```



The screenshot shows the Microsoft Azure Data Studio interface with the following details:

- Title Bar:** File, Edit, View, Help, Table_Creation.sql - ptd00000-sql-server.database.windows.net.cs-dsa-4513-sql-db (ptd00000) - Azure Data Studio
- Toolbar:** Run, Cancel, Disconnect, Change Connection, cs-dsa-4513-sql-db, Explain, Enable SQLCMD
- Code Area:** The main area displays the following SQL script for creating tables:

```
113  
114 CREATE TABLE [Cut_Job]  
115 (  
116     [Job_number] INT Primary Key, -- Primary Key column  
117     [Type_machine_used] NVARCHAR(50) NOT NULL,  
118     [Amt_time_machine_used] TIME NOT NULL,  
119     [Material_used] NVARCHAR(50) NOT NULL,  
120     [Labor_time] TIME NOT NULL,  
121     [date_completion] DATE,  
122     FOREIGN KEY(Job_number) REFERENCES Job  
123 );  
124     Create Index a on Cut_Job(Job_Number)  
125  
126  
127 CREATE TABLE Cost_Transactions  
128 (  
129     [Transaction_number] INT NOT NULL PRIMARY KEY, -- Primary Key column  
130     [sup_cost] int NOT NULL  
131 );  
132  
133 CREATE TABLE [Records]  
134 (  
135     [Transaction_number] INT NOT NULL Primary Key,  
136     [Job_number] INT NOT NULL,  
137     FOREIGN KEY(Transaction_number) REFERENCES Cost_Transactions,  
138     FOREIGN KEY(Job_number) REFERENCES Job,  
139 );  
140 );  
141 CREATE INDEX b on Records(Transaction_number)  
142  
143 CREATE TABLE Updates  
144 (  
145     [Transaction_number] INT NOT NULL,  
146     [Account_no] int NOT NULL,  
147     FOREIGN KEY(Transaction_number) REFERENCES Cost_Transactions
```

The code is color-coded for syntax highlighting, and the interface includes standard Windows-style icons for file operations.

The screenshot shows the Azure Data Studio interface with several tabs open. The main tab contains the following SQL code:

```
142 CREATE TABLE [Updates]
143 (
144     [Transaction_number] INT NOT NULL,
145     [Account_no] int NOT NULL,
146     FOREIGN KEY([Transaction_number]) REFERENCES Cost_Transactions,
147     FOREIGN KEY([Account_no]) REFERENCES Account,
148     Primary Key([Transaction_number,Account_no])
149 );
150
151 CREATE TABLE [Fit_Process]
152 (
153     [Process_ID] varchar(25) Primary Key, -- Primary Key column
154     [Fit_type] varchar(25) NOT NULL,
155     FOREIGN KEY([Process_ID]) REFERENCES Process
156 );
157
158 CREATE TABLE [Paint_Process]
159 (
160     [Process_ID] varchar(25) Primary Key, -- Primary Key column
161     [Type] varchar(25) NOT NULL,
162     [Method] NUMERIC NOT NULL,
163     FOREIGN KEY([Process_ID]) REFERENCES Process
164 );
165
166 CREATE TABLE [Cut_Process]
167 (
168     [Process_ID] varchar(25) Primary Key, -- Primary Key column
169     [Cutting_Type] varchar(25) NOT NULL,
170     [Machine_type] varchar(25) NOT NULL,
171     FOREIGN KEY([Process_ID]) REFERENCES Process
172 );
173
174
175
```

The status bar at the bottom shows the connection details: potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db. The system tray indicates the date and time as 11/19/2019 9:18 PM.

Successfully created

Messages

9:20:18 PM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:00.883

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

y.sql - disconnected SQLQuery_1 - potd0000...td0000 Main_SQLsql - disconnected Create_Procedure_Input.sql - potd0000...td0000 Table_Creation.sql - disconnected ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

1 select * from Customer
2 select * from Assembly
3 select * from Process
4 select * from Department
5 select * from Fit_Process
6 select * from Paint_Process
7 select * from Cut_Process
8 select * from Assigned
9 select * from Job
10 select * from Fit_Job
11 select * from Paint_Job
12 select * from Cut_Job
13 select * from Records
14 select * from Cost_Transactions
15 select * from Updates
16 select * from Account
17 select * from Assembly_account
18 select * from Department_account
19 select * from Process_account
20
21

```

Results Messages

| | Name | Address | Category |
|---|---------|---------|----------|
| 1 | Akhil | Norman | 8 |
| 2 | Bhairav | NYC | 8 |
| 3 | Bryan | SFO | 6 |
| 4 | Jack | Chicago | 8 |

In 21, Col 1 (497 selected) Spaces: 4 UTF-8 CRLF SQL MSSQL 76 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG 11:36 PM US 11/19/2019

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

y.sql - disconnected SQLQuery_1 - potd0000...td0000 Main_SQLsql - disconnected Create_Procedure_Input.sql - potd0000...td0000 Table_Creation.sql - disconnected ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```
14 select * from Cost_Transactions
```

Results Messages

| | Name | Address | Category |
|---|---------|---------|----------|
| 1 | Akhil | Norman | 8 |
| 2 | Bhairav | NYC | 8 |
| 3 | Bryan | SFO | 6 |
| 4 | Jack | Chicago | 8 |
| 5 | Naveen | Norman | 1 |
| 6 | Ryan | OKC | 5 |
| 7 | Taras | Dallas | 3 |

| | Assembly_ID | Date_ordered | Assembly_details | Name |
|---|-------------|--------------|------------------------|--------|
| 1 | A1 | 2017-08-01 | Ordered a big assembly | Naveen |
| 2 | A2 | 2018-06-01 | To get car doors | Taras |
| 3 | A3 | 2018-08-01 | Smelting unit | Akhil |

| | Process_ID | Process_data | Dept_Number |
|---|------------|--------------------------|-------------|
| 1 | P1 | Cutting car doors | 101 |
| 2 | P2 | Glazing car doors | 201 |
| 3 | P3 | Creating moulds for cars | 301 |

In 21, Col 1 Spaces: 4 UTF-8 CRLF SQL MSSQL 76 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG 11:36 PM US 11/19/2019

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

y.sql - disconnected SQLQuery_1 - potd0000...td0000) Main_SQLsql - disconnected Create_Procedure_Input.sql - potd0000...td0000) Table_Creation.sql - disconnected ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

14 select * from Cost_Transactions

Results Messages

| Job_number | Assembly_ID | Process_ID |
|------------|-------------|------------|
| 1 | A1 | P2 |
| 2 | A1 | P1 |
| 3 | A1 | P3 |
| 4 | A2 | P1 |
| 5 | A2 | P2 |
| 6 | A3 | P3 |
| 7 | A3 | P1 |
| 8 | A3 | P2 |

| Job_number | date_commencement |
|------------|-------------------|
| 1 | 2018-10-01 |
| 2 | 2018-11-01 |
| 3 | 2018-12-01 |
| 4 | 2019-01-01 |
| 5 | 2019-02-01 |
| 6 | 2019-03-01 |
| 7 | 2019-03-01 |
| 8 | 2019-04-01 |

| Job_number | Labor_time | date_completion |
|------------|------------|-----------------|
| 1 | 01:30:00 | 2019-01-01 |
| 2 | 02:30:00 | 2019-02-01 |

| Job_number | Color | Volume | Labor_time | date_completion |
|------------|-------|--------|------------|-----------------|
| 1 | 2 | Green | 50 | 02:30:00 |
| 2 | 6 | Yellow | 50 | 03:00:00 |
| 3 | 8 | Red | 30 | 02:00:00 |

| Job_number | Type_machine_used | Amt_time_machine_used | Material_used | Labor_time | date_completion |
|------------|-------------------|-----------------------|---------------|-------------------|-----------------|
| 1 | 3 | Cutter | 02:50:00 | iron sheet | 02:30:00 |
| 2 | 4 | Cutting bumpers | 01:30:00 | Coated steel | 03:00:00 |
| 3 | 7 | Cutting tyres | 02:30:00 | Vulcanized Rubber | 02:50:00 |

| Transaction_number | Job_number |
|--------------------|------------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

In 21, Col 1 Spaces: 4 UTF-8 CRLF SQL MSSQL 76 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG US 11:36 PM 11/19/2019

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

y.sql - disconnected SQLQuery_1 - potd0000...td0000) Main_SQLsql - disconnected Create_Procedure_Input.sql - potd0000...td0000) Table_Creation.sql - disconnected ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

14 select * from Cost_Transactions

Results Messages

| | Transaction_number | sup_cost |
|---|--------------------|----------|
| 1 | 1 | 10 |
| 2 | 2 | 20 |
| 3 | 3 | 15 |

| | Transaction_number | Account_no |
|---|--------------------|------------|
| 1 | 1 | 1002 |
| 2 | 1 | 1003 |
| 3 | 1 | 1005 |
| 4 | 2 | 1004 |
| 5 | 2 | 1005 |
| 6 | 2 | 1006 |
| 7 | 3 | 1001 |
| 8 | 3 | 1004 |
| 9 | 3 | 1006 |

| | Account_no | Date_of_establishment |
|---|------------|-----------------------|
| 1 | 1001 | 2017-08-01 |
| 2 | 1002 | 2017-10-01 |

In 21, Col 1 Spaces: 4 UTF-8 CRLF SQL MSSQL 76 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG US 11/19/2019

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

y.sql - disconnected SQLQuery_1 - potd0000...td0000) Main_SQLsql - disconnected Create_Procedure_Input.sql - potd0000...td0000) Table_Creation.sql - disconnected ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

14 select * from Cost_Transactions

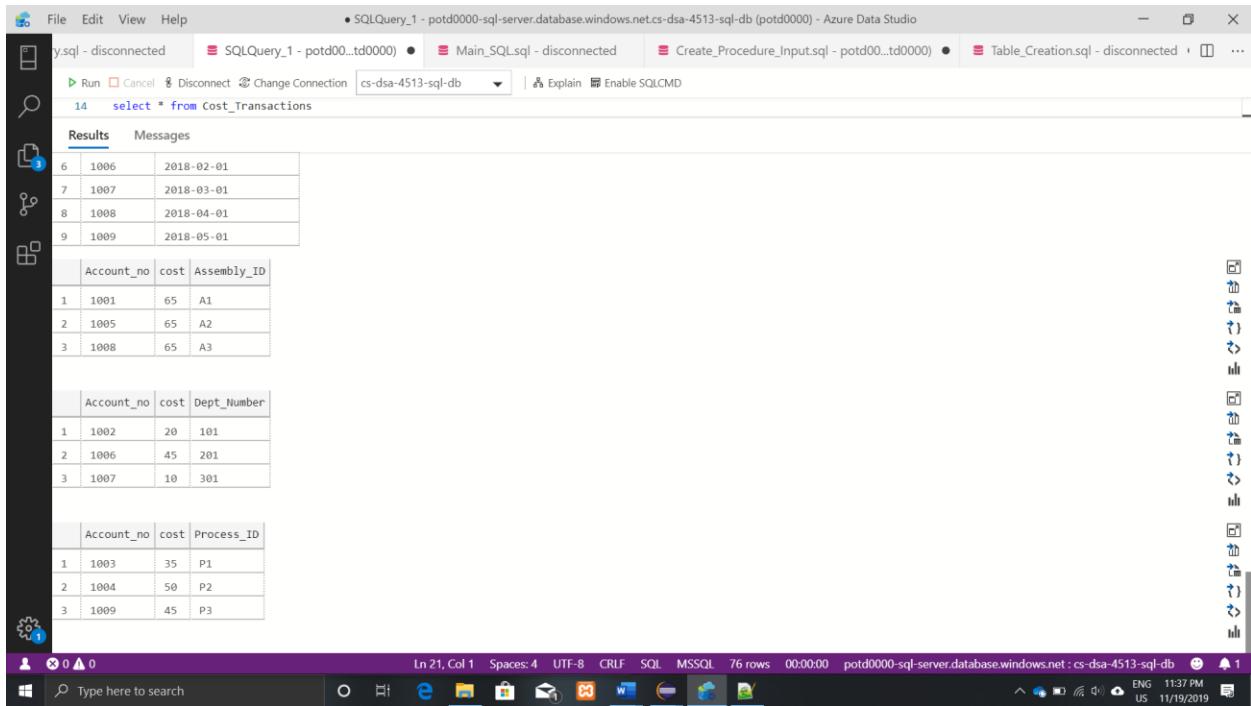
Results Messages

| | Account_no | Date_of_establishment |
|---|------------|-----------------------|
| 1 | 1001 | 2017-08-01 |
| 2 | 1002 | 2017-10-01 |
| 3 | 1003 | 2017-11-01 |
| 4 | 1004 | 2017-12-01 |
| 5 | 1005 | 2018-01-01 |
| 6 | 1006 | 2018-02-01 |
| 7 | 1007 | 2018-03-01 |
| 8 | 1008 | 2018-04-01 |
| 9 | 1009 | 2018-05-01 |

| | Account_no | cost | Assembly_ID |
|---|------------|------|-------------|
| 1 | 1001 | 65 | A1 |
| 2 | 1005 | 65 | A2 |
| 3 | 1008 | 65 | A3 |

| | Account_no | cost | Dept_Number |
|---|------------|------|-------------|
| 1 | 1002 | 20 | 101 |
| 2 | 1006 | 45 | 201 |
| 3 | 1007 | 10 | 301 |

In 21, Col 1 Spaces: 4 UTF-8 CRLF SQL MSSQL 76 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG US 11/19/2019



Task 5. The Java source program and screenshots showing its successful compilation

```

import java.sql.Connection;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.CallableStatement;
import java.util.Scanner;
import userdetails;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.Writer;

```

```

import java.nio.file.Files;
import java.nio.file.Paths;

public class Main
{
    public static void customer_entry(String name, String add, int cat)//Defining class for Procedure
1
    {
        String query = "{ call customer_entry(?, ?, ?) }"; //Calling procedure #1
        ResultSet rs;
        // Connect to database
        final String hostName = "potd0000-sql-server.database.windows.net";
        final String dbName = "cs-dsa-4513-sql-db";
        final String user = "potd0000";
        final String password = userdetails.password;
        final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
        try
        (final Connection connection = DriverManager.getConnection(url))
        {
            final String schema = connection.getSchema();
            System.out.println("Successful connection - Schema:" + schema);
            System.out.println("Inserting Customer:"+name);
            try
            (final Connection conn = DriverManager.getConnection(url);
             CallableStatement stmt = conn.prepareCall(query))
            {
                //Providing
parameters to the procedure

```

```

stmt.setString(1, name);
stmt.setString(2, add);
stmt.setInt(3, cat);
rs = stmt.executeQuery(); //Query execution
}
catch (SQLException ex)
{
    System.out.println(ex.getMessage());
}
}
catch (SQLException e)
{
    throw new RuntimeException(e);
}
}

public static void department_entry(int num, String data)//Defining class for Procedure 2
{
String query = "{ call department_entry(?,?) }"; //Calling procedure #2
ResultSet rs;
//Connect to database
final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;hostNameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
try

```

```

(final Connection connection = DriverManager.getConnection(url))
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Inserting Department number: "+num);
    try
    (final Connection conn = DriverManager.getConnection(url);
        CallableStatement stmt = conn.prepareCall(query))
    {
        //Providing parameters to the procedure
        stmt.setInt(1, num);
        stmt.setString(2, data);
        rs = stmt.executeQuery();                                //Query execution
    }
    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }
}
catch (SQLException e)
{
    throw new RuntimeException(e);
}
}

public static void assembly_entry(String id, String date, String details, String name)
{
    String query = "{ call assembly_entry(?, ?, ?, ?) }"; //Calling procedure #3
}

```

```

ResultSet rs;
//  

Connect to database  

final String hostName = "potd0000-sql-server.database.windows.net";  

final String dbName = "cs-dsa-4513-sql-db";  

final String user = "potd0000";  

final String password = userdetails.password;  

final String url =  

String.format("jdbc:sqlserver://%" + hostName + ";database=%s;user=%s;password=%s;encrypt=true;host  

NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);  

try  

(final Connection connection = DriverManager.getConnection(url))  

{  

final String schema = connection.getSchema();  

System.out.println("Successful connection - Schema:" + schema);  

System.out.println("Inserting Assembly details for customer: " + name);  

try  

(final Connection conn = DriverManager.getConnection(url);  

CallableStatement stmt = conn.prepareCall(query))  

{  

//Providing parameters to the procedure  

stmt.setString(1, id);  

stmt.setString(2, date);  

stmt.setString(3, details);  

stmt.setString(4, name);  

rs = stmt.executeQuery(); //Query execution  

}  

catch (SQLException ex)  

{

```

```

        System.out.println(ex.getMessage());
    }
}

catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

public static void process_entry(String id, String details, int num)
{
    String query = "{ call process_entry(?, ?, ?) }"; //Calling procedure #4
    ResultSet rs;
}

//Connect to database
final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
try
{
    final Connection connection = DriverManager.getConnection(url)
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Inserting Process details for department no.: "+num);
    try
    {
        final Connection conn = DriverManager.getConnection(url);
}
}
}

```

```

        CallableStatement stmt = conn.prepareCall(query))

    {

        //Providing parameters to the procedure
        stmt.setString(1, id);
        stmt.setString(2, details);
        stmt.setInt(3, num);
        rs = stmt.executeQuery();                                //Query execution

    }

    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }

}

catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

public static void account_entry(int num, String date, int ch, int cost, int id, String id2)
//Account parent entry

{
    String query = "{ call account_entry(?,?) }"; //Calling procedure #5

    ResultSet rs;
    ResultSet rs1;

    //Connect to database
    final String hostName = "potd0000-sql-server.database.windows.net";

```

```

final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try
{
    final Connection connection = DriverManager.getConnection(url)

    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Inserting Account details(parent) for acc.no: "+num);
    try(final Connection conn = DriverManager.getConnection(url);

        CallableStatement stmt = conn.prepareCall(query))

    {
        //Providing parameters to the procedure
        stmt.setInt(1, num);
        stmt.setString(2, date);
        rs = stmt.executeQuery();                                //Query execution
    }

    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }

    System.out.println("Inserting Account details(child) for acc.no: "+num);
    if (ch==3)
    {
        String query1= "{ call department_account_entry(?, ?, ?) }"; //Calling procedure #5
        try(final Connection conn = DriverManager.getConnection(url);

```

```

CallableStatement stmt1 = conn.prepareCall(query1))

{

//Providing parameters to the procedure

    stmt1.setInt(1, num);

    stmt1.setInt(2, cost);

    stmt1.setInt(3, id);

    rs = stmt1.executeQuery();           //Query execution

}

catch (SQLException ex)

{

    System.out.println(ex.getMessage());

}

}

else if(ch == 1)

{

    String query2 = "{ call assembly_account_entry(?, ?, ?) }"; //Calling procedure #5

    try(final Connection conn = DriverManager.getConnection(url);

        CallableStatement stmt2 = conn.prepareCall(query2))

    {

//Providing parameters to the procedure

        stmt2.setInt(1, num);

        stmt2.setInt(2, cost);

        stmt2.setString(3, id2);

        rs1 = stmt2.executeQuery();           //Query execution

    }

    catch (SQLException ex)

{

```

```

        System.out.println(ex.getMessage());
    }
}

else
{
    String query2 = "{ call process_account_entry(?, ?, ?) }"; //Calling procedure #5
    try(final Connection conn = DriverManager.getConnection(url);
        CallableStatement stmt2 = conn.prepareCall(query2))
    {

        //Providing parameters to the procedure
        stmt2.setInt(1, num);
        stmt2.setInt(2, cost);
        stmt2.setString(3, id2);
        rs1 = stmt2.executeQuery(); //Query execution
    }

    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }
}

}

catch (SQLException e)
{
    throw new RuntimeException(e);
}
}

```

```

public static void job_entry(int num, String date, String a_id, String p_id)
{
    String query = "{ call job_entry(?, ?, ?, ?) }"; //Calling procedure #4

    ResultSet rs;
}

//Connect to database

final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%;s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try
{
    (final Connection connection = DriverManager.getConnection(url))

    {
        final String schema = connection.getSchema();
        System.out.println("Successful connection - Schema:" + schema);
        System.out.println("Inserting Job details for job no.: "+num);
        try
        {
            (final Connection conn = DriverManager.getConnection(url);

            CallableStatement stmt = conn.prepareCall(query))

            {

                //Providing parameters to the procedure
                stmt.setInt(1, num);
                stmt.setString(2, date);
                stmt.setString(3, a_id);
                stmt.setString(4, p_id);
                rs = stmt.executeQuery(); //Query execution
            }
        }
    }
}

```

```

        }

        catch (SQLException ex)

        {

            System.out.println(ex.getMessage());

        }

    }

    catch (SQLException e)

    {

        throw new RuntimeException(e);

    }

}

public static void fit_job_completion(int num, String l_time, String date)

{

    String query = "{ call fit_job_completion(?, ?, ?) }"; //Calling procedure #4

    ResultSet rs;

    //Connect to database

    final String hostName = "potd0000-sql-server.database.windows.net";

    final String dbName = "cs-dsa-4513-sql-db";

    final String user = "potd0000";

    final String password = userdetails.password;

    final String url =

String.format("jdbc:sqlserver://%" + hostName + ";database=%s;user=%s;password=%s;encrypt=true;hostNameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

    try

    (final Connection connection = DriverManager.getConnection(url))

    {

        final String schema = connection.getSchema();

        System.out.println("Successful connection - Schema:" + schema);

    }

}

```

```

System.out.println("Inserting Job details for fit job.: ");
try
(final Connection conn = DriverManager.getConnection(url);
 CallableStatement stmt = conn.prepareCall(query))
{
    //Providing parameters to the procedure
    stmt.setInt(1, num);
    stmt.setString(2, l_time);
    stmt.setString(3, date);
    rs = stmt.executeQuery();                                //Query execution
}
catch (SQLException ex)
{
    System.out.println(ex.getMessage());
}
catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

public static void paint_job_completion(int num, String col, int vol, String l_time, String date)
{
    String query = "{ call paint_job_completion(?, ?, ?, ?, ?) }"; //Calling procedure #4
    ResultSet rs;
}

//Connect to database

```

```

final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%" + hostName + ";database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try
(final Connection connection = DriverManager.getConnection(url))
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Inserting Job details for paint job");
    try
(final Connection conn = DriverManager.getConnection(url);
    CallableStatement stmt = conn.prepareCall(query))
{
    //Providing parameters to the procedure
    stmt.setInt(1, num);
    stmt.setString(2, col);
    stmt.setInt(3, vol);
    stmt.setString(4, l_time);
    stmt.setString(5, date);
    rs = stmt.executeQuery(); //Query execution
}
catch (SQLException ex)
{
    System.out.println(ex.getMessage());
}
}

```

```

        }

    catch (SQLException e)

    {

        throw new RuntimeException(e);

    }

}

public static void cut_job_completion(int num, String typ, String a_time, String mat, String l_time, String date)

{

    String query = "{ call cut_job_completion(?, ?, ?, ?, ?, ?) }"; //Calling procedure #4

    ResultSet rs;

    //

Connect to database

    final String hostName = "potd0000-sql-server.database.windows.net";

    final String dbName = "cs-dsa-4513-sql-db";

    final String user = "potd0000";

    final String password = userdetails.password;

    final String url =

String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host

NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

    try

    (final Connection connection = DriverManager.getConnection(url))

    {

        final String schema = connection.getSchema();

        System.out.println("Successful connection - Schema:" + schema);

        System.out.println("Inserting Job details for cut job");

        try

        (final Connection conn = DriverManager.getConnection(url);

            CallableStatement stmt = conn.prepareCall(query))

```

```

{
    //Providing parameters to the procedure
    stmt.setInt(1, num);
    stmt.setString(2, typ);
    stmt.setString(3, a_time);
    stmt.setString(4, mat);
    stmt.setString(5, l_time);
    stmt.setString(6, date);
    rs = stmt.executeQuery();                                //Query execution
}

catch (SQLException ex)
{
    System.out.println(ex.getMessage());
}

}

catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

public static void updates_entry(int t_num, int s_cost, int j_num)
{
    String query = "{ call cost_transaction_record_entry(?, ?, ?) }"; //Calling procedure #8 1st
    part
    String query1 = "{ call updates_entry(?) }"; //Calling procedure #8 Part 2
    ResultSet rs;
}

Connect to database
//
```

```

final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%" + hostName + ";database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try
(final Connection connection = DriverManager.getConnection(url))
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Inserting Transaction details for job");
    try
    (final Connection conn = DriverManager.getConnection(url);
     CallableStatement stmt = conn.prepareCall(query))
    {
        //Providing parameters to the procedure sub part 1
        stmt.setInt(1, t_num);
        stmt.setInt(2, s_cost);                                // Taking entry for transactions
        stmt.setInt(3, j_num);
        rs = stmt.executeQuery();                            //Query execution
    }
    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }
    try
    (final Connection conn = DriverManager.getConnection(url);

```

```

CallableStatement stmt1 = conn.prepareCall(query1))

{

    //Providing parameters to the procedure sub part 2
    stmt1.setInt(1, t_num);

    rs = stmt1.executeQuery();                                //Query execution to update cost

}

catch (SQLException ex)

{
    System.out.println(ex.getMessage());
}

}

catch (SQLException e)

{
    throw new RuntimeException(e);
}

}

public static void cost_assembly_retrieve(String a_id)

{
    String query = "{ call cost_assembly_retrieve(?) }"; //Calling procedure #9

    ResultSet rs;

    Connect to database

    final String hostName = "potd0000-sql-server.database.windows.net";
    final String dbName = "cs-dsa-4513-sql-db";
    final String user = "potd0000";
    final String password = userdetails.password;
}

```

```

final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try (final Connection connection = DriverManager.getConnection(url))

{

    final String schema = connection.getSchema();

    System.out.println("Successful connection - Schema:" + schema);

    System.out.println("=====");

    try

    (final Connection conn = DriverManager.getConnection(url);

        CallableStatement resultset = conn.prepareCall(query))

    {

        resultset.setString(1, a_id); // Taking input for query

        rs = resultset.executeQuery(); // Execution

        System.out.println("Retrieving the Cost for assembly_id: "+ a_id);

        while (rs.next())

        {

            //Displaying table from SQL database

            System.out.println(String.format("%f ",rs.getFloat(1))); //Printing the result

        }

    }

    catch (SQLException e)

    {

        throw new RuntimeException(e);

    }

}

catch (SQLException e1)

{
    throw new RuntimeException(e1);
}

```

```

}

}

public static void labor_time_retrieve(String date,int d_num)
{
    String query = "{ call labor_time_retrieve(?,?) }"; //Calling procedure #10
    ResultSet rs;
}

Connect to database

final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try (final Connection connection = DriverManager.getConnection(url))
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("=====");
    //final String selectSql = "SELECT * FROM Performer";

    try
    (final Connection conn = DriverManager.getConnection(url));
        CallableStatement resultset = conn.prepareCall(query))
    {
        resultset.setString(1, date); // Taking input for query
        resultset.setInt(2, d_num);
        rs = resultset.executeQuery();
    }
}

```

```

//resultSet.getInt(1);           // Execution
System.out.println("Retrieving the Total time for date: "
+ date + "and department number: "+d_num);
//System.out.println(String.format("%d ",rs.getInt(1)));      //Printing the result
while (rs.next())
{
    //Displaying table from SQL database
    System.out.println(String.format("%d ",rs.getInt(1)));      //Printing the result
}
}

catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

catch (SQLException e1)
{
    throw new RuntimeException(e1);
}

}

public static void job_retrieve(String date,int dnum,int typ)
{
    String query;
    ResultSet rs;
    //Connect to database
    final String hostName = "potd0000-sql-server.database.windows.net";
    final String dbName = "cs-dsa-4513-sql-db";

```

```

final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
try (final Connection connection = DriverManager.getConnection(url))

{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("=====");
    if (typ == 1)
    {
        query = "{ call fit_job_retrieve(?,?) }"; //Calling sub procedure for queryy 12
        try
        (final Connection conn = DriverManager.getConnection(url));
            CallableStatement resultset = conn.prepareCall(query))
        {
            resultset.setString(1, date); // Taking input for query
            resultset.setInt(2, dnum);
            rs = resultset.executeQuery(); // Execution
            System.out.println("Retrieving the job info and assembly_ids");
            while (rs.next())
            {
                //Displaying table from SQL database
                System.out.println(String.format("%d |%s |%s |%s |",
                    rs.getInt(1),
                    rs.getString(2),
                    rs.getString(3),
                    rs.getString(4)));
                //Printing the result
            }
        }
    }
}

```

```

    }

    catch (SQLException e)

    {

        throw new RuntimeException(e);

    }

}

else if (typ == 2)

{

query = "{ call paint_job_retrieve(?,?) }"; //Calling sub procedure for query 12

try

(final Connection conn = DriverManager.getConnection(url);

    CallableStatement resultset = conn.prepareCall(query))

{

    resultset.setString(1, date); // Taking input for query

    resultset.setInt(2, dnum);

    rs = resultset.executeQuery(); // Execution

    System.out.println("Retrieving the job info and assembly_ids");

    while (rs.next())

    {

        //Displaying table from SQL database

        System.out.println(String.format("%d |%s |%d |%s |%s |%s |%s |",

            rs.getInt(1),

            rs.getString(2),

            rs.getInt(3),

            rs.getString(4),

            rs.getString(5),

            rs.getString(6))); //Printing the result

    }

}

catch (SQLException e)

```

```

    {
        throw new RuntimeException(e);
    }
}

else
{
    query = "{ call cut_job_retrieve(?,?) }"; //Calling sub procedure for query 12
    try
    {
        final Connection conn = DriverManager.getConnection(url);

        CallableStatement resultset = conn.prepareCall(query))

        {

            resultset.setString(1, date); // Taking input for query
            resultset.setInt(2, dnum);
            rs = resultset.executeQuery(); // Execution
            System.out.println("Retrieving the job info and assembly_ids");
            while (rs.next())
            {
                //Displaying table from SQL database
                System.out.println(String.format("%d |%s |%s |%s |%s |%s |",
                    rs.getInt(1),
                    rs.getString(2),
                    rs.getString(3),
                    rs.getString(4),
                    rs.getString(5),
                    rs.getString(6)));
                //Printing the result
            }
        }
    }
    catch (SQLException e)
    {
        throw new RuntimeException(e);
    }
}

```

```

        }

    }

}

catch (SQLException e1)
{
    throw new RuntimeException(e1);
}

}

public static void customer_retrieve(int st,int end)
{
    String query = "{ call customer_retrieve(?,?) }"; //Calling procedure #9
    ResultSet rs;
    //Connect to database
    final String hostName = "potd0000-sql-server.database.windows.net";
    final String dbName = "cs-dsa-4513-sql-db";
    final String user = "potd0000";
    final String password =.userdetails.password;
    final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;hostNameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
    try (final Connection connection = DriverManager.getConnection(url))
    {
        final String schema = connection.getSchema();
        System.out.println("Successful connection - Schema:" + schema);
        System.out.println("=====");
        try
        (final Connection conn = DriverManager.getConnection(url);

```

```

CallableStatement resultset = conn.prepareCall(query))

{

    resultset.setInt(1,st);

    resultset.setInt(2,end); // Taking input for query

    rs = resultset.executeQuery(); // Execution

    System.out.println("Retrieving the customer names in the range: ");

    while (rs.next())

    {

        //Displaying table from SQL database

        System.out.println(String.format("%s |%s |%d |",

            rs.getString(1),

            rs.getString(2),

            rs.getInt(3))); //Printing the result

    }

}

catch (SQLException e)

{

    throw new RuntimeException(e);

}

}

catch (SQLException e1)

{

    throw new RuntimeException(e1);

}

}

public static void cut_job_deletion(int st, int end)

{

    String query = "{ call cut_job_deletion(?,?) }"; //Calling procedure #4

```

```

        ResultSet rs;

        // Connect to database

final String hostName = "potd0000-sql-server.database.windows.net";
final String dbName = "cs-dsa-4513-sql-db";
final String user = "potd0000";
final String password = userdetails.password;
final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try
    (final Connection connection = DriverManager.getConnection(url))
{
    final String schema = connection.getSchema();
    System.out.println("Successful connection - Schema:" + schema);
    System.out.println("Deleting cut job: ");
    try
        (final Connection conn = DriverManager.getConnection(url);
         CallableStatement stmt = conn.prepareCall(query))
    {
        //Providing parameters to the procedure
        stmt.setInt(1, st);
        stmt.setInt(2, end);
        rs = stmt.executeQuery();                                //Query execution
    }
    catch (SQLException ex)
    {
        System.out.println(ex.getMessage());
    }
}

```

```

        catch (SQLException e)
    {
        throw new RuntimeException(e);
    }

}

public static void color_change(String col, int j_num)
{
    String query = "{ call color_change(?,?) }"; //Calling procedure #4
    ResultSet rs;
    // Connect to database
    final String hostName = "potd0000-sql-server.database.windows.net";
    final String dbName = "cs-dsa-4513-sql-db";
    final String user = "potd0000";
    final String password = userdetails.password;
    final String url =
        String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
    try
    {
        (final Connection connection = DriverManager.getConnection(url))
        {
            final String schema = connection.getSchema();
            System.out.println("Successful connection - Schema:" + schema);
            System.out.println("Change the color of job no.: "+j_num);
            try
            {
                (final Connection conn = DriverManager.getConnection(url));
                CallableStatement stmt = conn.prepareCall(query))
                {
                    //Providing parameters to the procedure

```

```

stmt.setString(1, col);
stmt.setInt(2, j_num);
rs = stmt.executeQuery(); //Query execution
}

catch (SQLException ex)
{
    System.out.println(ex.getMessage());
}

}

catch (SQLException e)
{
    throw new RuntimeException(e);
}

}

public static void take_input(String filename)
{
    String csvFile = filename;
    BufferedReader br = null;
    String line = "";
    String cvsSplitBy = ",";

    try
    {
        br = new BufferedReader(new FileReader(csvFile));
        while ((line = br.readLine()) != null)
        {

```

```

// use comma as separator
String[] file = line.split(cvsSplitBy);
String name = file[0];
String add = file[1];
int cat = Integer.parseInt(file[2]);

customer_entry(name,add,cat);

}

}

catch (FileNotFoundException e)
{
    e.printStackTrace();
}

catch (IOException e)
{
    e.printStackTrace();
}

finally
{
    if (br != null)
    {
        try
        {
            br.close();
        } catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}

```

```

        }
    }
}

}

public static void export_customer(int st,int end) throws IOException
{
    String query = "{ call customer_retrieve(?,?) }"; //Calling procedure #9
    ResultSet rs;
    //Connect to database
    String trial = "C:/Users/akhil/eclipse-workspace/HW3/src/output.txt";
    final String hostName = "potd0000-sql-server.database.windows.net";
    final String dbName = "cs-dsa-4513-sql-db";
    final String user = "potd0000";
    final String password = userdetails.password;
    final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);
    FileWriter fstream = new FileWriter(trial);
    BufferedWriter out = new BufferedWriter(fstream);
    try (final Connection connection = DriverManager.getConnection(url))
    {
        final String schema = connection.getSchema();
        System.out.println("Successful connection - Schema:" + schema);
        System.out.println("=====");
        try
        (final Connection conn = DriverManager.getConnection(url));
            CallableStatement resultset = conn.prepareCall(query))
        {

```

```

        resultSet.setInt(1,st);

        resultSet.setInt(2,end); // Taking input for query

        rs = resultSet.executeQuery(); // Execution

        System.out.println("Retrieving the customer names in the range: ");

        while (rs.next()) {

            out.write(rs.getString(1) + ", ");

            out.write(rs.getString(2) + ", ");

            out.write(rs.getInt(3) + ", ");

            out.newLine();

            /*out.write(System.getProperty("line.separator"));*/

        }

        System.out.println("Completed writing into text file");

        out.close();

    }

    catch (SQLException e)

    {

        throw new RuntimeException(e);

    }

}

catch (SQLException e1)

{

    throw new RuntimeException(e1);

}

}

public static void retrieve_process(String a_id)

{

```

```

String query = "{ call retrieve_process(?) }"; //Calling procedure #9

ResultSet rs;

//Connect to database

final String hostName = "potd0000-sql-server.database.windows.net";

final String dbName = "cs-dsa-4513-sql-db";

final String user = "potd0000";

final String password = userdetails.password;

final String url =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;", hostName, dbName, user, password);

try (final Connection connection = DriverManager.getConnection(url))

{

    final String schema = connection.getSchema();

    System.out.println("Successful connection - Schema:" + schema);

    System.out.println("=====");

    try

    (final Connection conn = DriverManager.getConnection(url);

        CallableStatement resultset = conn.prepareCall(query))

    {

        resultset.setString(1, a_id); // Taking input for query

        rs = resultset.executeQuery(); // Execution

        System.out.println("Retrieving the Cost for assembly_id: " + a_id);

        while (rs.next())

        {

            //Displaying table from SQL database

            System.out.println(String.format("%s ", rs.getString(1))); //Printing the result

        }

    }

    catch (SQLException e)

    {

}

```

```

        throw new RuntimeException(e);
    }

}

catch (SQLException e1)
{
    throw new RuntimeException(e1);
}

}

public static void main(String[] args) throws IOException
{
    String nl = System.getProperty("line.separator");           //Defining line separator
    System.out.println("1. Enter Customer information" + nl + "2. Enter Department details"
+ nl + "3. Entry Assembly table" + nl + "4. Enter Process table"
+ nl + "5. Enter Account information" + nl+ "6. Enter Job Information (Assigned Table) "
+ nl + "7. Enter Job Completion Data & Info"+nl + "8. Insert transaction info and update
accounts"
+nl+ "9. Retrieve cost for the assembly"+nl+ "10. Retrive total labor time for a given
date"
+nl+ "11. Retrieve the Process and related Department accessed by an assembly"
+nl+ "12. Retrieve a job information and the assembly id given a date and department"
+nl+ "13. Retrieve customer information for a category range"
+nl+ "14. Delete cut-jobs in a given job number range"
+nl+ "15. Change the color of a paint job"
+nl+ "16. Import new customer details from data file"
+nl+ "17. Export retrieved customer data in a category range"
+nl+ "18. Quit");
}

```

```

        while (true)
        {

            System.out.println("Please Enter Option Number: ");//Obtaining user input for
option

            Scanner user_input = new Scanner(System.in);

            int option = user_input.nextInt();

            if (option == 1)

            {

                System.out.println("Enter Customer Name: ");//Customer name as PK
                user_input.nextLine();

                String name = user_input.nextLine();

                System.out.println("Enter Customer Address: ");//Customer Address
entry

                String add = user_input.nextLine();

                System.out.println("Enter Customer Category: ");//Customer category
input

                int cat = user_input.nextInt();

                customer_entry(name, add, cat);//Providing input to procedure for
query 1

            }

            else if (option == 2)

            {

```

```

        System.out.println("Enter Department number: ");//Department
number as PK

        int d_num = user_input.nextInt();

        System.out.println("Enter information about the department:
");//Department info, about the work that goes on

        user_input.nextLine();

        String data = user_input.nextLine();

        department_entry(d_num, data);//Providing input to procedure 2

    }

    else if (option == 3)

    {

        System.out.println("Enter Assembly ID: ");//Assembly ID entry

        String id = user_input.next();

        System.out.println("Enter date of ordering: ");//Customer Address entry

        String date = user_input.next();

        System.out.println("Enter assembly details: ");//assembly details input

        user_input.nextLine();

        String details = user_input.nextLine();

        System.out.println("Enter Customer name for whom the assembly is
ordered: ");//Customer name as PK

        String name = user_input.next();

        assembly_entry(id, date, details, name);//Providing input to procedure
for query 1

    }

```

```

else if (option == 4)
{
    System.out.println("Enter Process ID:");//Process ID entry
    String id = user_input.next();

    System.out.println("Enter data about the process : ");//Process data
entry
    user_input.nextLine();
    String data = user_input.nextLine();

    System.out.println("Enter department number overseeing this process:
");//department input
    int num = user_input.nextInt();

    process_entry(id, data, num);//Providing input to procedure for query 1
}

else if (option == 5)
{
    System.out.println("Enter Account num (4 digit long): ");//Account
number
    int num = user_input.nextInt();

    System.out.println("Enter date of establishment for the acc : ");//Date
of establishment
    String date = user_input.next();

    System.out.println("Choose:" +nl+ "1. For assembly account"

```

```

+nl+ "2. For process account" +nl+ "3. For dept
account"+nl);// Choosing account type

        int ch = user_input.nextInt();

        int id;

        String id2;

        if (ch == 3)

        {

            System.out.println("Enter department number (3 digit long):

");//Dept. number entry

            id = user_input.nextInt();

            id2=null;

        }

        else

        {

            System.out.println("Enter Assembly/Process ID:

");//Assembly/Process number entry

            id = 0;

            id2=user_input.next();

        }

        System.out.println("Enter the cost: ");

        int cost = user_input.nextInt();

        account_entry(num, date, ch, cost, id, id2);//Providing input to

procedure for query 6

    }

}

else if (option == 6)

{

    System.out.println("Enter Job Number: ");//Account number

    int num = user_input.nextInt();

```

```

System.out.println("Enter date of commencement for the job : ");//Date
of establishment

String date = user_input.next();

System.out.println("Enter Assembly_ID for the Job:"+nl);// Choosing
account id

String a_id = user_input.next();

System.out.println("Enter Process ID for the Job:"+nl);// Choosing
process id

String p_id = user_input.next();

job_entry(num, date, a_id, p_id);//Providing input to procedure for
query 5

}

else if (option == 7)

{

System.out.println("Enter Job Number: ");//Account number

int num = user_input.nextInt();

System.out.println("Enter date of completion for the job : ");//Date of
establishment

String date = user_input.next();

System.out.println("Enter labor time for the Job (hr:min:sec):"+nl);//
Choosing account type

String l_time = user_input.next();

System.out.println("Choose the Job type:" +nl+ "1. Fit Job"
+nl+ "2. Paint Job" +nl+ "3. Cut Job");// Choosing
account type

int ch = user_input.nextInt();

```

```

        if(ch == 1)

        {

            fit_job_completion(num,l_time,date);//Providing input to
procedure for query 7 - fit job type

        }

        else if(ch == 2)

        {

            System.out.println("Enter color for the paint job :");//color
entry

            String col = user_input.next();

            System.out.println("Enter volume"+nl);// entering volume value
int vol = user_input.nextInt();

            paint_job_completion(num,col,vol,l_time,date);//Providing
input to procedure for query 7 - paint job type

        }

        else

        {

            System.out.println("Enter the type of machine used for the cut
job :");//type of machine entry

            String typ = user_input.next();

            System.out.println("Enter the material for the cut job :
");//material entry

            String mat = user_input.next();

            System.out.println("Enter amount of time machine was used for
(hr:min:sec)");// entering volume value

            String a_time = user_input.next();

```

```

        cut_job_completion(num,typ,a_time,mat,l_time,date);//Providing input to procedure for query
7 - paint job type

    }

}

else if (option == 8)

{



System.out.println("Enter Transaction number: ");//Transaction number
entry

int t_num = user_input.nextInt();



System.out.println("Enter Job number for the requisite transaction:
");//Job number entry

int j_num = user_input.nextInt();



System.out.println("Enter the supplementary cost for the
transaction");// Enter sup cost

int s_cost = user_input.nextInt();



updates_entry(t_num, s_cost, j_num);//Providing input to procedure
for query 8

}

else if (option == 9)

{

System.out.println("Enter Assembly ID: ");//Transaction number entry

String a_id = user_input.next();

cost_assembly_retrieve(a_id);

```

```

    }

else if (option == 10)

{

    System.out.println("Enter the date: ");//Transaction number entry

    String date = user_input.next();

    System.out.println("Enter the department number: ");//Transaction
number entry

    int d_num = user_input.nextInt();

    labor_time_retrieve(date,d_num);

    // yet for display to work

}

else if (option == 11)

{

    System.out.println("Enter the assembly id: ");//Date entry

    String a_id = user_input.next();

    retrieve_process(a_id);

}

else if (option == 12)

{

    System.out.println("Enter the date: ");//Date entry

    String date = user_input.next();

    System.out.println("Enter the department number: ");//Department
number entry

    int d_num = user_input.nextInt();

    System.out.println("Choose the type of job: "

        +nl+"1. Fit job" +nl+ "2.Paint Job"

        +nl+ "3. Cut Job"); //Type of job

entry

    int typ = user_input.nextInt();

    job_retrieve(date,d_num,typ);
}

```

```

    }

    else if (option == 13)

    {

        System.out.println("Enter start for the category range: ");// entry of
start and end range of category

        int st = user_input.nextInt();

        System.out.println("Enter end for the category range: ");

        int end = user_input.nextInt();

        customer_retrieve(st,end);

    }

    else if (option == 14)

    {

        System.out.println("Enter start for the job no. range: ");// entry of start
and end range of job no

        int st = user_input.nextInt();

        System.out.println("Enter end for the job no. range: ");

        int end = user_input.nextInt();

        cut_job_deletion(st,end);

    }

    else if (option == 15)

    {

        System.out.println("Enter the job no.");// entry of the job number and
paint choice

        int j_num = user_input.nextInt();

        System.out.println("Enter color of the paint to be changed into ");

        String col = user_input.next();

        color_change(col,j_num);

    }

    else if (option == 16)

```

```

{
    System.out.println("Enter the file name to import data");// entry of the
job number and paint choice

    String file = user_input.next();
    take_input(file);
    // C:/Users/akhil/eclipse-workspace/HW3/src/input.csv;

}

else if (option == 17)

{
    System.out.println("Enter start for the category range: ");// entry of
start and end range of category

    int st = user_input.nextInt();

    System.out.println("Enter end for the category range: ");

    int end = user_input.nextInt();

    export_customer(st,end);

    //export_data(int st, int end, String filename)

}

else if (option == 18)

{
    System.out.println("You have chosen to Quit. Have a great day ahead");

    System.exit(0);//Breaking while loop

}

System.out.println("1. Enter Customer information" + nl + "2. Enter Department
details"

+ nl + "3. Entry Assembly table" + nl + "4. Enter Process table"
+ nl + "5. Enter Account information" + nl+ "6. Enter Job
Information (Assigned Table) "
+ nl + "7. Enter Job Completion Data & Info"+nl + "8. Insert
transaction info and update accounts"
+nl+ "9. Retrieve cost for the assembly"+nl+ "10. Retrive total
labor time for a given date"

```

```
+nl+ "11. Retrieve the Process and related Department accessed  
by an assembly"  
+nl+ "12. Retrieve a job information and the assembly id given a  
date and department"  
+nl+ "13. Retrieve customer information for a category range"  
+nl+ "14. Delete cut-jobs in a given job number range"  
+nl+ "15. Change the color of a paint job"  
+nl+ "16. Import new customer details from data file"  
+nl+ "17. Export retrieved customer data in a category range"  
+nl+ "18. Quit");  
}  
}  
}
```

Starting from empty tables

eclipse-workspace - HW3/src/Mainjava - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Navigator (D...)

HW3

Main.java

Main_copy.java

src

Metadata

Settings

bin

userdetails.java

classpath

project

hs_err_pid12448.log

hs_err_pid13008.log

SampleAzureSQLProject

settings

bin

src

sample.java

userdetails.java

classpath

project

hs_err_pid19660.log

hs_err_pid6884.log

replay_pid6884.log

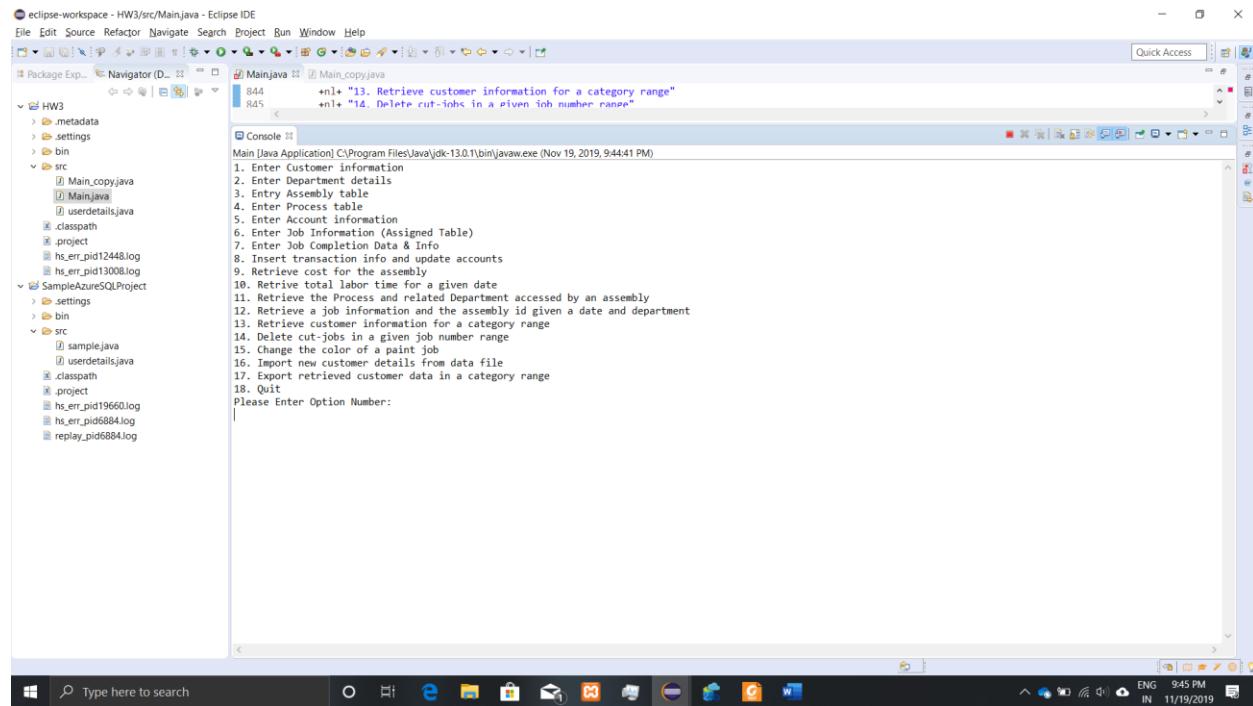
Main.java [Main_copy.java]

844 +nl+ "13. Retrieve customer information for a category range"
845 +nl+ "14. Delete cut-inhs in a given job number range"

Console

Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:44:41 PM)
1. Enter Customer Information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

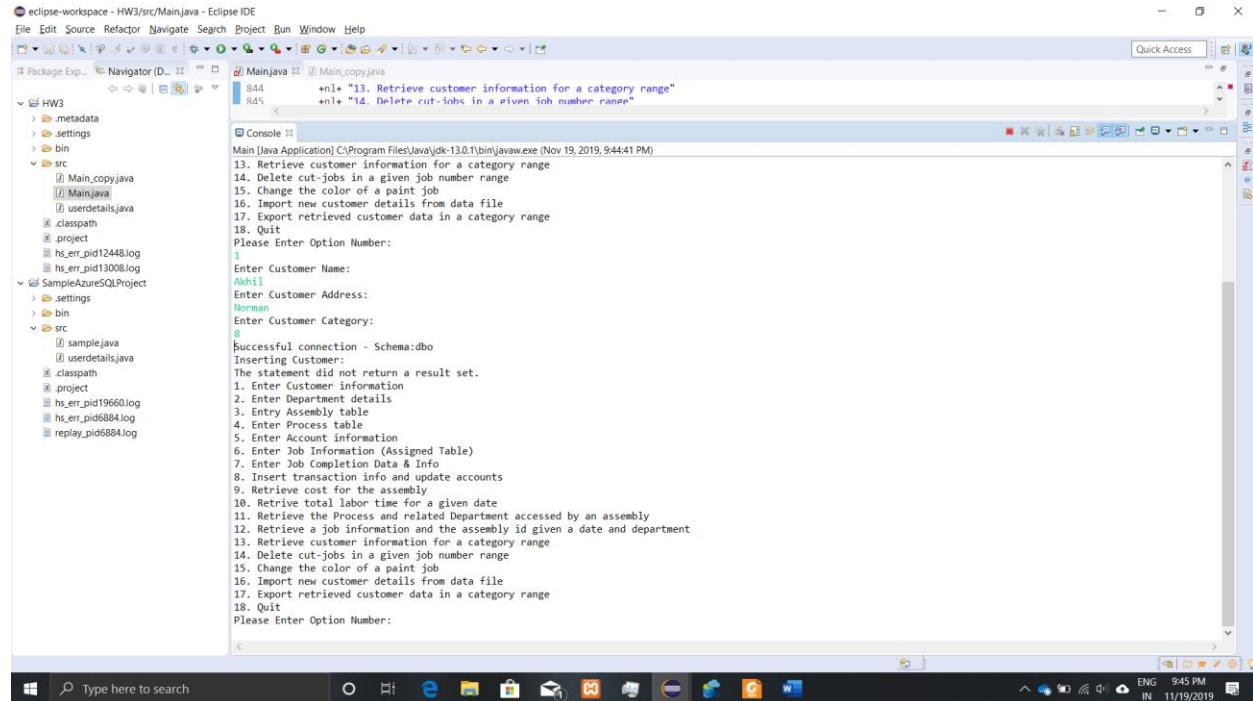
Task 6. Java program Execution



The screenshot shows the Eclipse IDE interface with a Java application named "Main.java" running in the console. The console output lists 18 numbered options for interacting with a database. The user has entered option 1, which prompts for customer information.

```
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:44:41 PM)
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Job Information
6. Enter Job Completion Data & Info
7. Insert transaction info and update accounts
8. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
```

Screenshots showing the testing of query 1



This screenshot shows the continuation of the Java application's execution. After selecting option 1, the user is prompted to enter customer information. The application successfully connects to the database and inserts a new customer record.

```
1
Enter Customer Name:
Akhil
Enter Customer Address:
Norman
Enter Customer Category:
8
Successful connection - Schema:dbo
Inserting Customer:
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account Information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
```

The screenshot shows the Azure Data Studio interface. In the top bar, there are several tabs: SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio, SQL_Query.sql - disconnected, Create_Procedure_Input.sql - potd00...td0000, SQLQuery_1 - potd00...td0000, Main_SQLsql - potd00...td0000, and Table_Creation.sq. The main area displays a query window with the following content:

```
1  SELECT * from Customer
2
```

The results pane shows a table with three columns: Name, Address, and Category. There is one row of data:

| | Name | Address | Category |
|---|-------|---------|----------|
| 1 | Akhil | Norman | 8 |

At the bottom of the screen, the Windows taskbar is visible with various icons.

Screenshots showing the testing of query 2

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The left sidebar shows the Package Explorer with projects HW3 and SampleAzureSQLProject. The Navigator view shows Main.java selected. The right side has a Console view and a Quick Access bar. The Java code in Main.java contains the following logic:

```
844    /*13. Retrieve customer information for a category range*/
845    /*14. Delete cut-ids in a given job number range*/
846
847    Main.java [D:\Main_copy.java]
848
849    844    /*13. Retrieve customer information for a category range*/
850    845    /*14. Delete cut-ids in a given job number range*/
851
852    Main.java [D:\Main_copy.java]
853
854    Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:47:47 PM)
855
856    11. Retrieve the Process and related Department accessed by an assembly
857    12. Retrieve a Job information and the assembly id given a date and department
858    13. Retrieve customer information for a category range
859    14. Delete cut-jobs in a given job number range
860    15. Change the color of a paint job
861    16. Import new customer details from data file
862    17. Export retrieved customer data in a category range
863    18. Quit
864
865    Please Enter Option Number:
866
867    2
868    Enter Department number:
869        101
870    Enter information about the department:
871        Soldering
872    Successful connection - Schema:dbo
873    Inserting Department number: 101
874
875    The statement did not return a result set.
876    1. Enter Department
877    2. Enter Department details
878    3. Entry Assembly table
879    4. Enter Process table
880    5. Enter Account information
881    6. Enter Job Information (Assigned Table)
882    7. Enter Job Completion Data & Info
883    8. Insert transaction info and update accounts
884    9. Retrieve cost for the assembly
885    10. Retrieve total labor time for a given date
886    11. Retrieve the Process and related Department accessed by an assembly
887    12. Retrieve a Job information and the assembly id given a date and department
888    13. Retrieve customer information for a category range
889    14. Delete cut-jobs in a given job number range
890    15. Change the color of a paint job
891    16. Import new customer details from data file
892    17. Export retrieved customer data in a category range
893    18. Quit
894
895    Please Enter Option Number:
```

At the bottom of the screen, the Windows taskbar is visible with various icons.

A screenshot of the Azure Data Studio interface. The title bar shows the connection details: SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio. The main area displays a query window with the following content:

```
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD
1  SELECT * from Department
2
```

The results pane shows a table with two columns: Dept_Number and Data. The data row is:

| Dept_Number | Data |
|-------------|-----------|
| 101 | Soldering |

The status bar at the bottom indicates the following information: 340 ▲ 0, Ln 1, Col 25, Spaces: 4, UTF-8, CRLF, SQL, MSSQL, 1 rows, 00:00:00, potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db, ENG 9:49 PM IN 11/19/2019.

Screenshots showing the testing of query 3

A screenshot of the Azure Data Studio interface. The title bar shows the connection details: SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio. The main area displays a query window with the following content:

```
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD
1  SELECT * from Assembly
2
```

The results pane shows a table with four columns: Assembly_ID, Date_ordered, Assembly_details, and Name. The data row is:

| Assembly_ID | Date_ordered | Assembly_details | Name |
|-------------|--------------|---------------------------|-------|
| A1 | 2017-07-01 | Ordering Multiple Assembl | Akhil |

The status bar at the bottom indicates the following information: 340 ▲ 0, Ln 1, Col 23, Spaces: 4, UTF-8, CRLF, SQL, MSSQL, 1 rows, 00:00:00, potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db, ENG 9:52 PM IN 11/19/2019.

eclipse-workspace - HW3/src/Main.java - Eclipse IDE

```

Main.java [3] [Main_copy.java]
844  +n1+ "13. Retrieve customer information for a category range"
845  +n1+ "14. Delete cut-jobs in a given job number range"

Console [1]
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:51:33 PM)
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
3
Enter Assembly ID:
A1
Enter date of ordering:
07-01-17
Enter assembly details:
Ordering Multiple Assemblies
Enter Customer name for whom the assembly is ordered:
Akhil
Successful connection - Schema:dbo
Inserting Assembly details for customer: Akhil
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

Screenshots showing the testing of query 4

eclipse-workspace - HW3/src/Main.java - Eclipse IDE

```

Main.java [3] [Main_copy.java]
844  +n1+ "13. Retrieve customer information for a category range"
845  +n1+ "14. Delete cut-jobs in a given job number range"

Console [1]
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:51:33 PM)
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
4
Enter Process ID:
P1
Enter data about the process :
Cutting Car doors
Enter department number overseeing this process:
101
Successful connection - Schema:dbo
Inserting Process details for department no.: 101
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

A screenshot of the Azure Data Studio interface. The top navigation bar includes File, Edit, View, Help, and a tab for SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio. Below the navigation bar, there are several tabs: SQL_Query.sql - disconnected, Create_Procedure_Input.sql - potd00...td0000, SQLQuery_1 - potd00...td0000, Main_SQLsql - potd00...td0000, and Table_Creation.sq. The main area shows a query editor with the following code:

```
1  SELECT * from Process
```

The Results tab is selected, displaying the following table:

| | Process_ID | Process_data | Dept_Number |
|---|------------|-------------------|-------------|
| 1 | P1 | Cutting Car doors | 101 |

The bottom status bar shows the connection details: potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db. The taskbar at the bottom includes icons for Start, Task View, Edge, File Explorer, Mail, Photos, Control Panel, Taskbar settings, File History, and Windows Update, along with system status indicators like battery level (340), network, and volume.

Screenshots showing the testing of query 5

eclipse-workspace - HW3/src/Mainjava - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Navigator (D...)

HW3

- src
- Main_copy.java
- Main.java
- userdetails.java
- classpath
- project
- hs_err_pid12448.log
- hs_err_pid13008.log
- hs_err_pid19660.log
- hs_err_pid6884.log
- replay_pid6884.log

Main.java [3] Main_copy.java

```

844     +n1+ "13. Retrieve customer information for a category range"
845     +n1+ "14. Delete cut-inhs in a given ioh number range"

```

Console

```

Main Java Application C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:55:14 PM)
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
5
Enter Account num (4 digit long):
1001
Enter date of establishment for the acc :
07-01-17
Choose:
1. For assembly account
2. For process account
3. For dept account
1
Enter Assembly/Process ID:
A1
Enter the cost:
50
Successful connection - Schema:dbo
Inserting Account details(parent) for acc.no: 1001
The statement did not return a result set.
Inserting Account details(child) for acc.no: 1001
The statement did not return a result set.
1. Enter Customer Information
2. Enter Customer details
3. Enter Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range

```

Type here to search

File Edit View Help

SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

SQL_Query.sql - disconnected Create_Procedure_Input.sql - potd00...td0000 SQLQuery_1 - potd00...td0000 Main_SQLsql - potd00...td0000 Table_Creation.sq ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

1 SELECT * from Account
2 SELECT * from Assembly_account
3

```

Results Messages

| Account_no | Date_of_establish... |
|------------|----------------------|
| 1001 | 2017-07-01 |

| Account_no | cost | Assembly_ID |
|------------|------|-------------|
| 1001 | 50 | A1 |

340 0 Type here to search

Screenshots showing the testing of query 6

eclipse-workspace - HW3/src/Main.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Navigator (D...)

HW3

- metadata
- settings
- bin
- src
 - Main_copy.java
 - Main.java**
 - userdetails.java
- classpath
- project
- hs_err_pid12448.log
- hs_err_pid13008.log

SampleAzureSQLProject

- settings
- bin
- src
 - sample.java
 - userdetails.java
- classpath
- project
- hs_err_pid19660.log
- hs_err_pid6884.log
- replay_pid6884.log

Main.java [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:59:04 PM)

```

844    /*1. Enter Customer information
845    +nl+ "13. Retrieve customer information for a category range"
846    +nl+ "14. Delete cut-inhs in a given job number range"

```

Console

```

Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:59:04 PM)
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
6
Enter Job Number:
1
Enter date of commencement for the job :
08-01-18
Enter Assembly_ID for the Job:
A1
Enter Process ID for the Job:
P1

```

Successful connection - Schema:dbo
Inserting Job details for job no.: 1
Violation of PRIMARY KEY constraint 'PK__Job__FB7534909E27DEAB'. Cannot insert duplicate key in object 'dbo.Job'. The duplicate key value is (1).
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table

Type here to search

File Edit View Help

SQLQuery_1 - potd0000-sql-server.database.windows.net:cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

SQL_Query.sql - disconnected Create_Procedure_Input.sql - potd00...td0000 SQLQuery_1 - potd00...td0000 Main_SQL.sql - potd00...td0000 Table_Creation.sq ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

1 Select * from Assigned

Results Messages

| Job_number | Assembly_ID | Process_ID |
|------------|-------------|------------|
| 1 | A1 | P1 |

Type here to search

In 1, Col 23 Spaces: 4 UTF-8 CRLF SQL MSSQL 1 rows 00:00:00 potd0000-sql-server.database.windows.net:cs-dsa-4513-sql-db ENG 10:00 PM IN 11/19/2019

Screenshots showing the testing of query 7

eclipse-workspace - HW3/src/Mainjava - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Navigator (D...)

HW3

- metadata
- settings
- bin
- src

 - Main_copy.java
 - Main.java
 - userdetails.java
 - classpath
 - project
 - hs_err_pid12448.log
 - hs_err_pid13008.log

- SampleAzureSQLProject

 - settings
 - bin
 - src

 - sample.java
 - userdetails.java
 - classpath
 - project
 - hs_err_pid19660.log
 - hs_err_pid6884.log
 - replay_pid6884.log

Main.java 844 845

```
+n1+ "13. Retrieve customer information for a category range"
+nl+ "14. Delete cut-jobs in a given job number range"
```

Console

Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:59:04 PM)
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
7
Enter Job Number:
1
Enter date of completion for the job :
01-01-2019
Enter labor time for the Job (hr:min:sec):
1:30
Choose the Job type:
1. Fit Job
2. Paint Job
3. Cut Job
1
Successful connection Scheme:dbo
Inserting Job details for fit job.
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost of an assembly
10. Retrieve total labor time for a given date
11. Retrieve all Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit

Type here to search

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio

SQL_Query.sql - disconnected Create_Procedure_Input.sql - potd00...td0000) SQLQuery_1 - potd00...td0000 Main_SQLsql - potd00...td0000 Table_Creation.sq ...

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

1 Select * from Fit_Job

Results Messages

| Job_number | Labor_time | date_completion |
|------------|------------|-----------------|
| 1 | 01:30:00 | 2019-01-01 |

340 0 Type here to search

Ln 1, Col 22 Spaces: 4 UTF-8 CRLF SQL MSSQL 1 rows 00:00:00 potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db ENG 10:03 PM IN 11/19/2019

Screenshots showing the testing of query 8

eclipse-workspace - HW3/src/MainJava - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer Navigator (D...) MainJava Main_copyjava

HW3

- .metadata
- settings
- bin
- sri
- Main_copy.java
- Main.java
- userdetails.java
- classpath
- project
- hs_err_pid12448.log
- hs_err_pid13008.log

SampleAzureSQLProject

- settings
- bin
- sri
- sample.java
- userdetails.java
- classpath
- project
- hs_err_pid19660.log
- hs_err_pid8844.log
- replay_pid6884.log

Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 9:59:04 PM)

14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
8
Enter Transaction number:
1
Enter Job number for the requisite transaction:
Enter the supplementary cost for the transaction:
20
Successful connection - Schema:dbo
Inserting Transaction details for job
The statement did not return a result set.
Cannot insert the value NULL into column 'Account_no', table 'cs-dsa-4513-sql-db.dbo.Updates'; column does not allow nulls. INSERT fails.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Insert transaction info (Assigned Table)
7. Enter Job Completion Date & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

Type here to search

10:12 PM ENG IN 11/19/2019

The screenshot shows the Azure Data Studio interface with three result grids displayed:

- Cost_Transactions:**

| | Transaction_number | sup_cost |
|---|--------------------|----------|
| 1 | 1 | 20 |
- Records:**

| | Transaction_number | Job_number |
|---|--------------------|------------|
| 1 | 1 | 1 |
- Updates:**

| | Transaction_number | Account_no |
|---|--------------------|------------|
| 1 | 1 | 1001 |

```

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio
SQL_Query.sql - disconnected Create_Procedure_Input.sql - potd00...td0000) ● SQLQuery_1 - potd00...td0000) ● Main_SQLsql - potd00...td0000) Table_Creation.sq ...
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD
1 select * from Assembly_account
2 select * from Process_account
3 select * from Department_account

Results Messages
Account_no | cost | Assembly_ID |
1          | 70   | A1

Account_no | cost | Process_ID |
Account_no | cost | Dept_Number |

```

Only the Assembly account is updated and not the others as it's the only one present for this example.

Screenshots showing the testing of query 9

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Navigator (D...) Main.java [Main_copy.java]
src Main_copy.java Main.java userdetails.java classpath project hs_err.pid12448.log hs_err.pid13008.log
HW3 metadata settings bin Main_copy.java Main.java userdetails.java classpath project hs_err.pid19660.log hs_err.pid6884.log replay.pid6884.log
SampleAzureSQLProject settings bin sample.java userdetails.java classpath project hs_err.pid19660.log hs_err.pid6884.log replay.pid6884.log

Console Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\java.exe (Nov 19, 2019, 9:59:04 PM)
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Insert new customer details from data file
17. Export retrieved customer data in a category range
18. Quit

Please Enter Option Number:
9
Enter Assembly ID:
A1
Successful connection - Schema:dbo
=====
Retrieving the Cost for assembly_id: A1
70.000000
1. Enter Customer information
2. Enter Department details
3. Enter Assembly table
4. Enter Job table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Insert new customer details from data file
17. Export retrieved customer data in a category range
18. Quit

Please Enter Option Number:

```

Screenshots showing the testing of query 10

```

eclipse-workspace - HW3/src/Mainjava - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:32:13 PM)
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
10
Enter the date:
01-01-19
Enter the department number:
101
Successful connection - Schema:dbo
=====
Retrieving the Total time for date: 01-01-19and department number: 101
0
1. Enter Customer Information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

Query runs in SQL

```

File Edit View Help • SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio
SQL_Query.sql - disconnected Create_Procedure_Input.sql - potd00...td0000 SQLQuery_1 - potd00...td0000 Main_SQL.sql - potd00...td0000 Table_Creation.sq ...
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD
1 go
2 exec labor_time_retrieve @end='01-01-2019',@dept=101

Messages
10:33:11 PM Started_executing_query_at_Line_2
90

Total execution time: 00:00:00.060


```

Screenshots showing the testing of query 11

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 11:40:43 PM)
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
11
Enter the assembly id:
A2
Successful connection - Schema:dbo
=====
Retrieving the Cost for assembly_id: A2
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

Screenshots showing the testing of query 12

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:23:59 PM)
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
12
Enter the date:
01-01-19
Enter the department number:
01
Choose the type of job:
1. Fit job
2. Paint Job
3. Cut Job
1
Successful connection - Schema:dbo
=====
Retrieving the job info and assembly_ids
1 [01:30:00.000000 | 2019-01-01 | A1 |
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

Screenshots showing the testing of query 13

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:37:15 PM)
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
13
Enter start for the category range:
4
Enter end for the category range:
9
Successful connection - Schema:dbo
=====Retrieving the customer names in the range:
Akhil [Nov 19, 2019 | 8:15 PM]
Brijesh [Nov 19, 2019 | 8:15 PM]
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit

```

Screenshots showing the testing of query 14

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:39:01 PM)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
14
Enter start for the job no. range:
4
Enter end for the job no. range:
7
Successful connection - Schema:dbo
Deleting cut job:
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

Screenshots showing the testing of query 15

Screenshot of Azure Data Studio showing the results of a SQL query:

```
1 select * from Paint_Job
```

| | Job_number | Color | Volume | Labor_time | date_completion |
|---|------------|--------|--------|------------|-----------------|
| 1 | 2 | Green | 50 | 02:30:00 | 2019-02-01 |
| 2 | 6 | Yellow | 50 | 04:00:00 | 2019-04-01 |
| 3 | 8 | Red | 30 | 02:00:00 | 2019-05-01 |

Screenshot of Eclipse IDE showing a Java application console output:

```
eclipse-workspace - HW3/src/Main.java - Eclipse IDE
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:40:26 PM)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
18
Enter the job no.
2
Enter color of the paint to be changed into
Black
Successful connection - Schema:dbo
Change the color of job no.: 2
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Enter Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
```

A screenshot of the Azure Data Studio interface. The top navigation bar shows 'File', 'Edit', 'View', 'Help' and the title 'SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio'. Below the title, there are tabs for 'SQL_Query.sql - disconnected', 'Create_Procedure_Input.sql - potd00...td0000', 'SQLQuery_1 - potd00...td0000', 'Main_SQLsql - potd00...td0000', and 'Table_Creation.sq'. A toolbar below the tabs includes 'Run', 'Cancel', 'Disconnect', 'Change Connection', 'cs-dsa-4513-sql-db', 'Explain', and 'Enable SQLCMD'. The main area contains a code editor with the following SQL query:

```
1  select * from Paint_Job
```

The results pane displays a table with the following data:

| | Job_number | Color | Volume | Labor_time | date_completion |
|---|------------|--------|--------|------------|-----------------|
| 1 | 2 | Black | 50 | 02:30:00 | 2019-02-01 |
| 2 | 6 | Yellow | 50 | 04:00:00 | 2019-04-01 |
| 3 | 8 | Red | 30 | 02:00:00 | 2019-05-01 |

The bottom status bar shows 'Ln 1, Col 24', 'Spaces: 4', 'UTF-8', 'CRLF', 'SQL', 'MSSQL', '3 rows', '00:00:00', 'potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db', and system information like 'ENG 10:40 PM IN 11/19/2019'.

Screenshots showing the testing of the import and export options

Import

A screenshot of the Eclipse IDE interface. The top menu bar shows 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', 'Help'. The title bar says 'eclipse-workspace - HW3/src/Main.java - Eclipse IDE'. The central workspace shows a 'Console' tab with the following text output:

```
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:43:10 PM)
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
16
Enter the file name to import data
C:/users/akhil/eclipse-workspace/HW3/src/input.csv
Successful connection - Schema:dbo
Inserting Customer:Aang
The statement did not return a result set.
Successful connection - Schema:dbo
Inserting Customer:Katarra
The statement did not return a result set.
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
```

The bottom status bar shows 'LN 1, COL 24', 'Spaces: 4', 'UTF-8', 'CRLF', 'SQL', 'MSSQL', '3 rows', '00:00:00', 'potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db', and system information like 'ENG 10:43 PM IN 11/19/2019'.

The screenshot shows the Azure Data Studio interface. In the top bar, there are several tabs: SQLQuery_1 - potd0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (potd0000) - Azure Data Studio, SQL_Query.sql - disconnected, Create_Procedure_Input.sql - potd00...td0000, SQLQuery_1 - potd00...td0000, Main_SQLsql - potd00...td0000, and Table_Creation.sq. Below the tabs, a toolbar includes Run, Cancel, Disconnect, Change Connection, Explain, and Enable SQLCMD. A code editor window contains the following SQL query:

```
1   select * from Customer
```

Below the code editor is a results grid titled "Results". The grid has columns: Name, Address, and Category. The data is as follows:

| | Name | Address | Category |
|---|--------|---------------------|----------|
| 1 | Aang | Southern Air Temple | 9 |
| 2 | Akhil | Norman | 8 |
| 3 | Bryan | SFO | 6 |
| 4 | Jack | Chicago | 8 |
| 5 | Katara | Northen Water Tribe | 7 |
| 6 | Naveen | Norman | 1 |
| 7 | Ryan | OKC | 5 |
| 8 | Taras | Dallas | 3 |

The status bar at the bottom of the screen shows: 340 ▲ 0, Ln 1, Col 23, Spaces: 4, UTF-8, CRLF, SQL, MSSQL, 8 rows, 00:00:00, potd0000-sql-server.database.windows.net : cs-dsa-4513-sql-db, ENG 10:43 PM IN 11/19/2019.

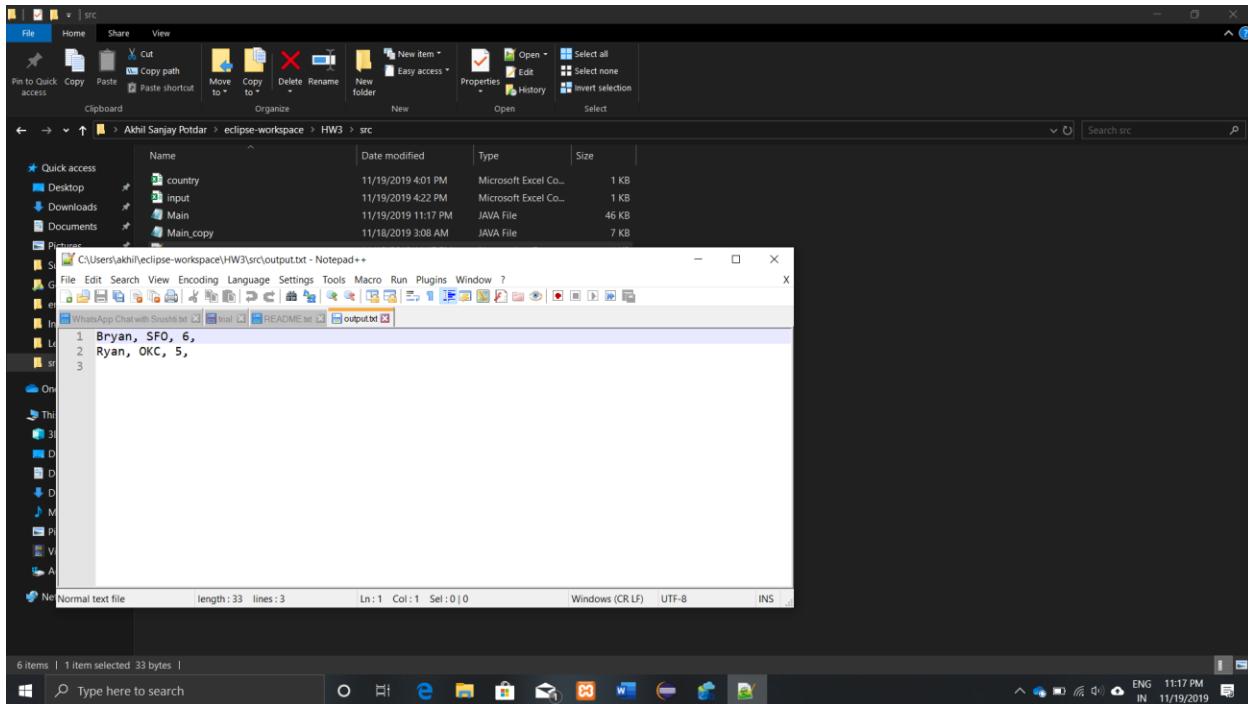
Export

The screenshot shows the Eclipse IDE interface. The title bar says "eclipse-workspace - HW3/src/Main.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The main area is the "Console" tab, which displays the following Java application output:

```
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 11:17:05 PM)
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
17
Enter start for the category range:
4
Enter end for the category range:
7
successful connection - Schema:db
=====
Retrieving the customer names in the range:
Completed writing into text file
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a even job number range
```

The status bar at the bottom of the screen shows: Type here to search, ENG 11:17 PM IN 11/19/2019.

Output



Screenshots showing the testing of three types of errors

1. Primary Key violation as data already present

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:43:10 PM)
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
1
Enter Customer Name:
Akhil
Enter Customer Address:
Norman
Enter Customer Category:
8
Successful connection - Schema:dbo
Inserting Customer-Akhil
Violation of PRIMARY KEY constraint 'PK__Customer__737584F71589205F'. Cannot insert duplicate key in object 'dbo.Customer'. The duplicate key value is (Akhil).
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:

```

2. Category input out of range

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:43:10 PM)
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
1
Enter Customer Name:
Toto
Enter Customer Address:
Hono
Enter Customer Category:
1111
Successful connection - Schema:dbo
Inserting Customer:Toto
The INSERT statement conflicted with the CHECK constraint "CK__Customer_Catego_62C64559". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.Customer", column 'Category'.
1. Enter Customer Information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit.
Please Enter Option Number:

```

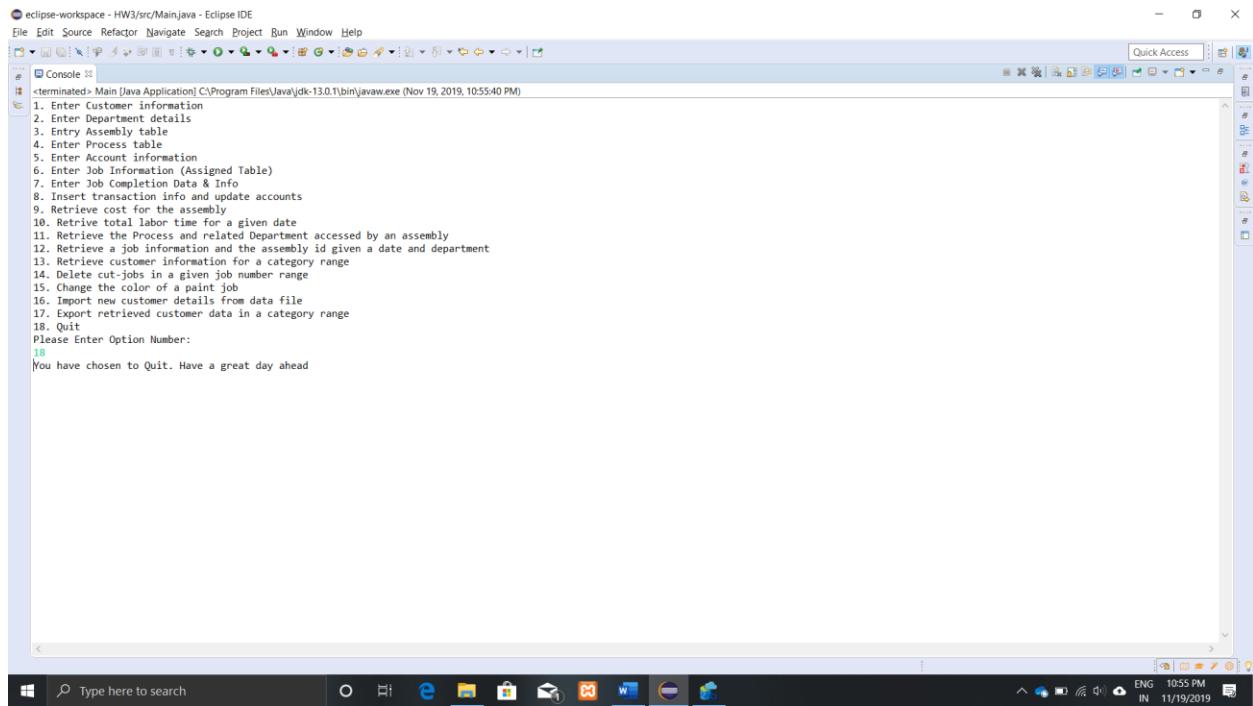
3. Foreign Key not present hence no input available

```

eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:47:18 PM)
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
0
Enter Job Number:
1
Enter date of commencement for the job :
01-01-29
Enter Assembly_ID for the Job:
A67
Enter Process ID for the Job:
P32
Successful connection - Schema:dbo
Inserting Job details for job no.: 1
Violation of PRIMARY KEY constraint 'PK__Job__FB7534909B833906'. Cannot insert duplicate key in object 'dbo.Job'. The duplicate key value is (1).
1. Enter Customer Information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Insert transaction info and update accounts
9. Retrieve cost for the assembly
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit.
Please Enter Option Number:

```

Screenshots showing the testing of the quit option



```
eclipse-workspace - HW3/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console [terminated] Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Nov 19, 2019, 10:55:40 PM)
1. Enter Customer information
2. Enter Department details
3. Entry Assembly table
4. Enter Process table
5. Enter Account information
6. Enter Job Information (Assigned Table)
7. Enter Job Completion Data & Info
8. Import transaction info and update accounts
9. Retrieve data from database
10. Retrieve total labor time for a given date
11. Retrieve the Process and related Department accessed by an assembly
12. Retrieve a Job information and the assembly id given a date and department
13. Retrieve customer information for a category range
14. Delete cut-jobs in a given job number range
15. Change the color of a paint job
16. Import new customer details from data file
17. Export retrieved customer data in a category range
18. Quit
Please Enter Option Number:
18
You have chosen to Quit. Have a great day ahead
```

Task 7. Web database application and its execution

Web database application source program and screenshots showing Its successful compilation

```
package ip_task7_potdar_akhil;
```

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DataHandler {

    private Connection conn;

    // Azure SQL connection credentials
    private String server = "potd0000-sql-server.database.windows.net";
```

```

private String database = "cs-dsa-4513-sql-db";
private String username = "potd0000";
private String password = "150%isSrushti$Singh";

// Resulting connection string
final private String url = String.format(
    "jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertifica
    te=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
    server, database, username, password);

// Initialize and save the database connection
private void getDBConnection() throws SQLException {
    if (conn != null) {
        return;
    }

    this.conn = DriverManager.getConnection(url);
}

// Return the result of selecting everything from the movie_night table
public ResultSet displayall() throws SQLException {
    getDBConnection();
    final String sqlQuery = "SELECT * FROM Customer;";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
    return stmt.executeQuery();
}

// Inserts a record into the movie_night table with the given attribute values

```

```

public ResultSet customer_retrieve(int st, int end) throws SQLException
{
    ResultSet rs;
    getDBConnection(); // Prepare the database connection

    // Prepare the SQL statement
    final String sqlQuery ="{ call customer_retrieve(?,?) }";
    final PreparedStatement resultset = conn.prepareCall(sqlQuery);

    // Replace the '?' in the above statement with the given attribute values
    resultset.setInt(1,st);
    resultset.setInt(2,end); // Taking input for query
    //rs = resultset.executeQuery(); // Execution

    // Execute the query, if only one record is updated, then we indicate success by
    returning true
    return resultset.executeQuery();
}

public boolean customer_entry(String name, String add, int cat) throws SQLException
{
    ResultSet rs;
    getDBConnection(); // Prepare the database connection

    // Prepare the SQL statement
    final String sqlQuery ="{ call customer_entry(?, ?, ?) }";
    final PreparedStatement resultset = conn.prepareCall(sqlQuery);

    // Replace the '?' in the above statement with the given attribute values
    resultset.setString(1, name);

```

```

        resultset.setString(2, add);
        resultset.setInt(3, cat);
        //rs = resultset.executeQuery(); // Execution

        // Execute the query, if only one record is updated, then we indicate success by
        returning true
        return resultset.executeUpdate() == 1;
    }

}

```

Character Retrieve Code (to display)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Customers</title>
  </head>
  <body>
    <%@page import="ip_task7_potdar_akhil.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
        // We instantiate the data handler here, and get all the movies from the
        database
        final DataHandler handler = new DataHandler();
        // Get the attribute values passed from the input form.
        String start = request.getParameter("start_range");
        String end = request.getParameter("end_range");

        /*
         * If the user hasn't filled out all the time, movie name and duration.
        This is very simple checking.
        */
        if (start.equals("") || end.equals(""))
        {
            response.sendRedirect("add_movie_form.jsp");
        }
    %>

```

```

else
{
    int start_1 = Integer.parseInt(start);
    int end_1 = Integer.parseInt(end);

    final ResultSet movies = handler.customer_retrieve(start_1,
end_1);

    %>
    <!-- The table for displaying all the movie records -->
    <table cellspacing="2" cellpadding="2" border="1">
        <tr> <!-- The table headers row -->
            <td align="center">
                <h4>Name</h4>
            </td>
            <td align="center">
                <h4>Address</h4>
            </td>
            <td align="center">
                <h4>Category</h4>
            </td>
        </tr>
    <%
        while(movies.next())
    {
        // For each movie_night record returned...
        // Extract the attribute values for every row
        returned
    }
    final String name = movies.getString("Name");
    final String add = movies.getString("Address");
    final int cat = movies.getInt("Category");

    out.println("<tr>"); // Start printing out the new
table row
    out.println( // Print each attribute value
        "<td align=\"center\">" + name +
        "</td><td align=\"center\"> " + add +
        "</td><td align=\"center\"> " + cat + "</td>");
    out.println("</tr>");
}
%>

<a href="get_all_customers.jsp">See all the customer present.</a>
<%
}
%>
</table>
</body>
</html>

```

Code to insert range for selection

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Customer Retrieval from Range</title>
  </head>
  <body>
    <h2>Customer Retrieval from Range</h2>
    <!--
      Form for collecting user input for the new movie_night record.
      Upon form submission, add_movie.jsp file will be invoked.
    -->
    <form action="character_retrieve.jsp">
      <!-- The form organized in an HTML table for better clarity. -->
      <table border=1>
        <tr>
          <th colspan="2">Enter the Category Range:</th>
        </tr>
        <tr>
          <td>Customer Range Start:</td>
          <td><div style="text-align: center;">
            <input type=text name=start_range>
          </div></td>
        </tr>
        <tr>
          <td>Customer Range End:</td>
          <td><div style="text-align: center;">
            <input type=text name=end_range>
          </div></td>
        </tr>
        <tr>
          <td><div style="text-align: center;">
            <input type=reset value=Clear>
          </div></td>
          <td><div style="text-align: center;">
            <input type=submit value=Submit>
          </div></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Code to display all customers

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Customers</title>
```

```

</head>
<body>
    <h2>Customer Table </h2>
    <%@page import="ip_task7_potdar_akhil.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
        // We instantiate the data handler here, and get all the movies from the
        database
        final DataHandler handler = new DataHandler();
        final ResultSet movies = handler.displayall();
    %>

    <!-- The table for displaying all the movie records -->
    <table cellspacing="2" cellpadding="2" border="1">
        <tr> <!-- The table headers row -->
            <td align="center">
                <h4>Name</h4>
            </td>
            <td align="center">
                <h4>Address</h4>
            </td>
            <td align="center">
                <h4>Category</h4>
            </td>
        </tr>

        <%
            while(movies.next()) { // For each movie_night record returned...
                // Extract the attribute values for every row returned
                final String name = movies.getString("Name");
                final String add = movies.getString("Address");
                final int cat = movies.getInt("Category");

                out.println("<tr>"); // Start printing out the new table row
                out.println( // Print each attribute value
                    "<td align=\"center\">" + name +
                    "</td><td align=\"center\"> " + add +
                    "</td><td align=\"center\"> " + cat + "</td>");
                out.println("</tr>");
            }
        %>
    </table>
    <ul>
        <a href="add_customer_info.jsp">Insert a new customer.</a>
    </ul>
    <ul>
        <a href="insert_range.jsp">Retrieve customer in a categorical range.</a>
    </ul>

</body>
</html>

```

Code to add customers

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
<%@page import="ip_task7_potdar_akhil.DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandler handler = new DataHandler();

// Get the attribute values passed from the input form.
String name = request.getParameter("Name");
String add = request.getParameter("Address");
String cat = request.getParameter("Category");

/*
 * If the user hasn't filled out all the time, movie name and duration. This is
very simple checking.
 */
if (name.equals("") || add.equals("") || cat.equals("")) {
    response.sendRedirect("add_customer_info.jsp");
} else {
    int category = Integer.parseInt(cat);

    // Now perform the query with the data from the form.
    boolean success = handler.customer_entry(name, add, category);
    if (!success) { // Something went wrong
        %>
            <h2>There was a problem inserting the course</h2>
        <%
    } else { // Confirm success to the user
        %>
            <h2>The New Customer Information:</h2>

            <ul>
                <li>Name: <%= name %></li>
                <li>Address: <%= add %></li>
                <li>Category: <%= category %></li>
            </ul>

            <h2>Was successfully inserted.</h2>

            <a href="get_all_customers.jsp">See all the customer present.</a>
        <%
    }
}
%>

```

```
</body>
</html>
```

Code to add information permanently into db

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Customer Entry</title>
    </head>
    <body>
        <h2>Customer Entry</h2>
        <!--
            Form for collecting user input for the new movie_night record.
            Upon form submission, add_movie.jsp file will be invoked.
        -->
        <form action="add_customer.jsp">
            <!-- The form organized in an HTML table for better clarity. -->
            <table border=1>
                <tr>
                    <th colspan="2">Enter the Customer Information:</th>
                </tr>
                <tr>
                    <td>Name:</td>
                    <td><div style="text-align: center;">
                        <input type=text name=Name>
                    </div></td>
                </tr>
                <tr>
                    <td>Address:</td>
                    <td><div style="text-align: center;">
                        <input type=text name=Address>
                    </div></td>
                </tr>
                <tr>
                    <td>Category:</td>
                    <td><div style="text-align: center;">
                        <input type=text name=Category>
                    </div></td>
                </tr>
                <tr>
                    <td><div style="text-align: center;">
                        <input type=reset value=Clear>
                    </div></td>
                    <td><div style="text-align: center;">
                        <input type=submit value=Insert>
                    </div></td></tr>
                </table>
            </form>
        </body>
    </html>
```

Project - ip_task7_ipotdar_akhil/src/ip_task7_ipotdar_akhil/DataHandler.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

```
1 package ip_task7_ipotdar_akhil;
2
3 import java.sql.Connection;
4
5 public class DataHandler {
6
7     private Connection conn;
8
9     // Azure SQL connection credentials
10    private String server = "potd0000-sql-server.database.windows.net";
11    private String database = "cs-dsa-4513-sql-db";
12    private String username = "potd0000";
13    private String password = "1508isSruShuti$Singh";
14
15    // Resulting connection string
16    final private String url = String.format(
17        "jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=%s.database.windows.net;loginTimeout=30",
18        server, database, username, password);
19
20    // Initialize and save the database connection
21    private void getDBConnection() throws SQLException {
22        if (conn != null) {
23            return;
24        }
25
26        this.conn = DriverManager.getConnection(url);
27    }
28
29    // Return the result of selecting everything from the movie_night table
30    public ResultSet displayAll() throws SQLException {
31        getDBConnection();
32        final String sqlQuery = "SELECT * FROM Customer";
33        final PreparedStatement stat = conn.prepareStatement(sqlQuery);
34        return stat.executeQuery();
35    }
36
37    // Inserts a record into the movie_night table with the given attribute values
38    public ResultSet customer_retrieve(int st, int end) throws SQLException {
39    }
40
41    // Prepare the SQL statement
42    public ResultSet customer_retrieve(int st, int end) throws SQLException {
43    }
44    {
45        ResultSet rs;
46        getDBConnection(); // Prepare the database connection
47
48        // Prepare the SQL statement
49    }
50
51    // Writable | Smart Insert | 72:57:2604 |
```

Project - ip_task7_pdotar_akhil/src/ip_task7_pdotar_akhil/DataHandler.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

```
36     final String sqlQuery = "SELECT * FROM Customer";
37     final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
38     return stmt.executeQuery();
39 }
40
41 // Inserts a record into the movie_night table with the given attribute values
42 public ResultSet customer_retrieve(int st, int end) throws SQLException
43 {
44     ResultSet rs;
45     getDBConnection(); // Prepare the database connection
46
47     // Prepare the SQL statement
48     final String sqlQuery = "{ call customer_retrieve(?,?) }";
49     final PreparedStatement resultset = conn.prepareCall(sqlQuery);
50
51     // Replace the '?' in the above statement with the given attribute values
52     resultset.setInt(1,st);
53     resultset.setInt(2,end); // Taking input for query
54     //rs = resultset.executeQuery(); // Execution
55
56     // Execute the query, if only one record is updated, then we indicate success by returning true
57     return resultset.executeQuery();
58 }
59 public boolean customer_entry(String name, String add, int cat) throws SQLException
60 {
61     ResultSet rs;
62     getDBConnection(); // Prepare the database connection
63
64     // Prepare the SQL statement
65     final String sqlQuery = "{ call customer_entry(?, ?, ?) }";
66     final PreparedStatement resultset = conn.prepareCall(sqlQuery);
67
68     // Replace the '?' in the above statement with the given attribute values
69     resultset.setString(1, name);
70     resultset.setString(2, add);
71     resultset.setInt(3, cat);
72     //rs = resultset.executeQuery(); // Execution
73
74     // Execute the query, if only one record is updated, then we indicate success by returning true
75     return resultset.executeUpdate() == 1;
76 }
77
78 }
```

Screenshots showing the testing of the Web database application

Screenshot of Eclipse IDE showing the Customer Table page.

Project Explorer:

- ip_task7_potdar_akhil
 - Deployment Descriptor: ip_task7_potdar_akhil
 - JAX-WS Web Services
 - Java Resources
 - JavaScript Resources
 - Referenced Libraries
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - add_customer_info.jsp
 - add_customer.jsp
 - character_retrieve.jsp
 - get_all_customers.jsp
 - insert_range.jsp
- ip_azure_test
- Servers

Customer Table:

| Name | Address | Category |
|--------|---------|----------|
| Akhil | Norman | 8 |
| Bryan | SFO | 6 |
| Jack | Chicago | 8 |
| Naveen | Norman | 1 |
| Ryan | OKC | 5 |
| Taras | Dallas | 3 |

[Insert a new customer.](#)
[Retrieve customer in a categorical range.](#)

Console:

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_231\bin\java.exe (Nov 19, 2019, 10:57:12 PM)
Nov 19, 2019 10:57:14 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARS that were scanned but
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Nov 19, 2019 10:57:14 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [708] milliseconds
```

Screenshot of Eclipse IDE showing the Customer Entry page.

Project Explorer:

- ip_task7_potdar_akhil
 - Deployment Descriptor: ip_task7_potdar_akhil
 - JAX-WS Web Services
 - Java Resources
 - JavaScript Resources
 - Referenced Libraries
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - add_customer_info.jsp
 - add_customer.jsp
 - character_retrieve.jsp
 - get_all_customers.jsp
 - insert_range.jsp
- ip_azure_test
- Servers

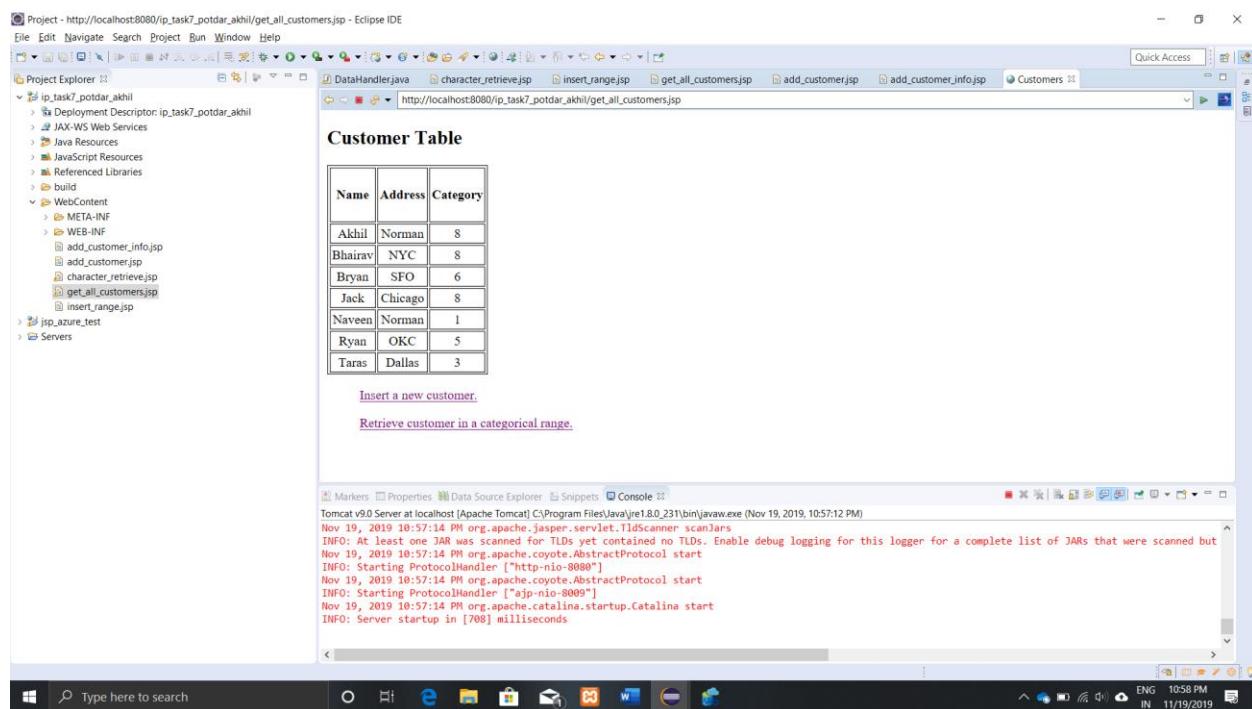
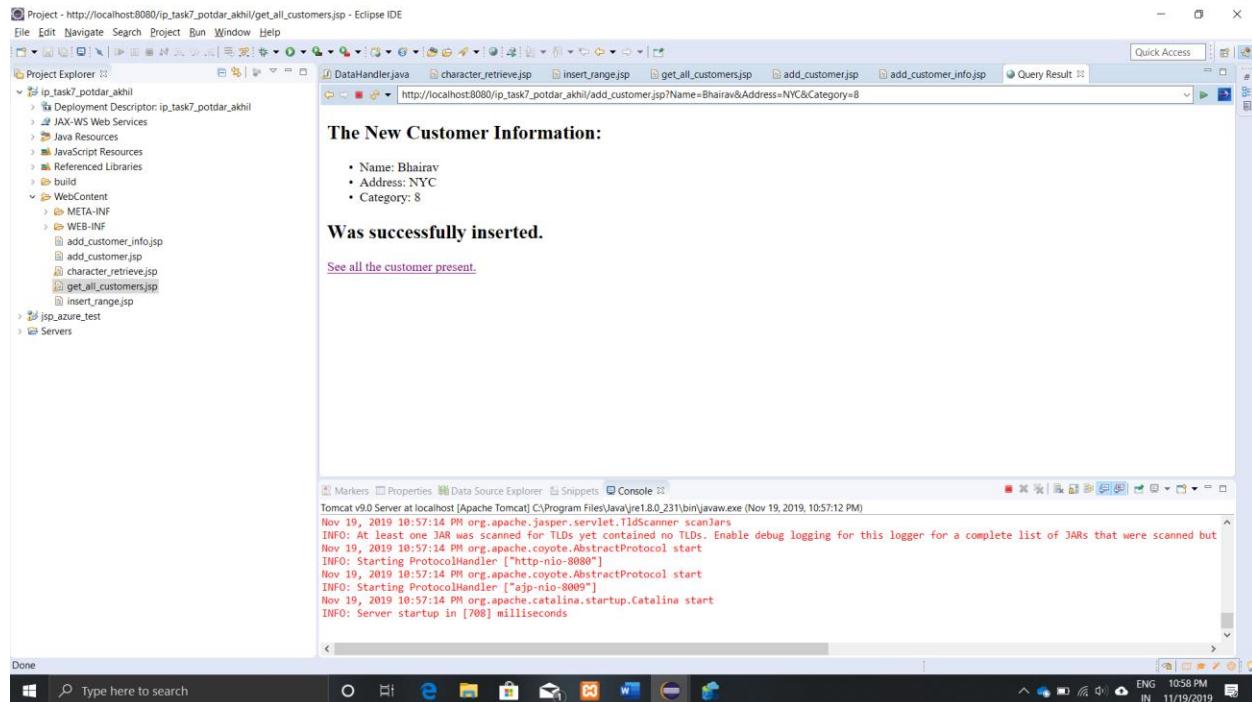
Customer Entry:

Enter the Customer Information:

| | |
|--------------------------------------|---------------------------------------|
| Name: | Bhairav |
| Address: | NYC |
| Category: | 8 |
| <input type="button" value="Clear"/> | <input type="button" value="Insert"/> |

Console:

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_231\bin\java.exe (Nov 19, 2019, 10:57:12 PM)
Nov 19, 2019 10:57:14 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARS that were scanned but
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Nov 19, 2019 10:57:14 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [708] milliseconds
```



Screenshot of Eclipse IDE showing a Java web application project named "ip_task7_potdar_akhil". The "Customer Retrieval from Range" page is displayed, showing a form to enter category range values (4 and 7) and a table of customer data.

Customer Retrieval from Range

| Enter the Category Range: | | |
|---------------------------|---|---------------------------------------|
| Customer Range Start: | 4 | <input type="button" value="Clear"/> |
| Customer Range End: | 7 | <input type="button" value="Submit"/> |

Customer Data Table

| Name | Address | Category |
|-------|---------|----------|
| Bryan | SFO | 6 |
| Ryan | OKC | 5 |

Console Output

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Nov 19, 2019, 10:57:12 PM)
Nov 19, 2019 10:57:14 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARS that were scanned but
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 19, 2019 10:57:14 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Nov 19, 2019 10:57:14 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [708] milliseconds
```