

Title: INDIVIDUAL PROJECT - A JOB-SHOP ACCOUNTING SYSTEM

Course: CS-4513 (DBMS)

Section Number: 001

Sem & Year: Fall 2021

Instructor: Dr. Le Gruenwald

Author: Ashesh Gaur

Author ID: 113462106

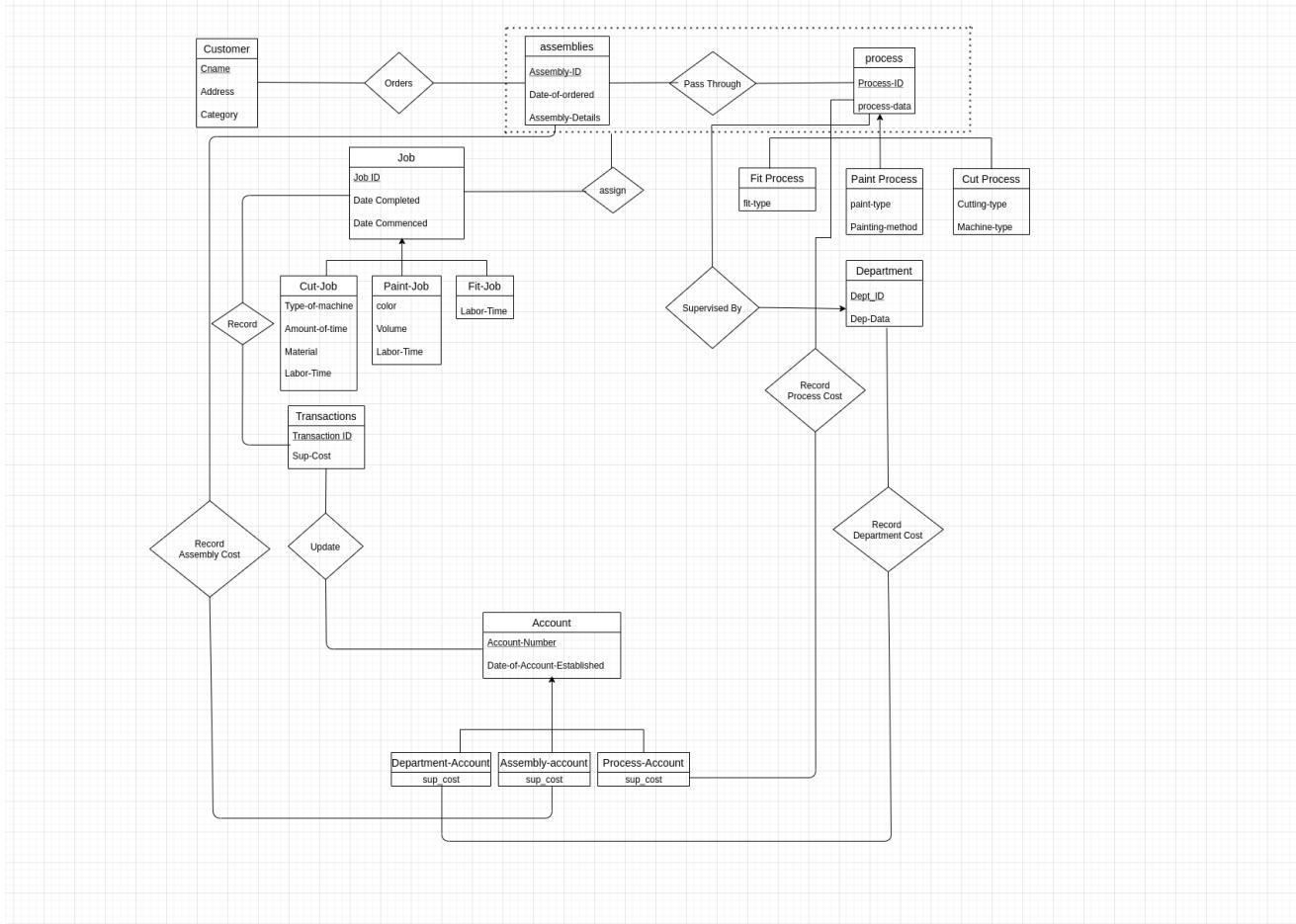
Email: ashesh.gaur-1@ou.edu

Course: CS-4513 (DBMS).....	1
Task 1	3
1.1 ER Diagram.....	3
1.2 Relational Schema	4
Task 2. Data Dictionary	5
Task 3	9
3.1 Discussion of Storage Structures for Tables	9
3.2 Discussion of storage structures for tables (Azure SQL Database)	12
Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database	12
Task 5	15
5.1 SQL statements (and Transact SQL stored procedures, if any) Implementing all queries (1-15 and error checking).....	15
Primary Key Constraint error check by Azure SQL Database	17
Foreign Key Constraint error check by Azure SQL Database:.....	18
Invalid query check by Azure SQL Database.....	18
CHECK Constraint check by Azure SQL Database	19
5.2 The Java source program and screenshots showing its successful compilation	19
Task 6. Java program Execution.....	30
Five queries for Type 1:	30
Five queries of Type 2	35
Ten queries for Type 3:.....	40
Ten Queries of Type 4:.....	59
Ten Queries of Type 5:.....	74
Ten Queries for Type 6:	84
Ten Queries for Type 7:	94
Ten Queries of type 8:	104
Three Queries of Type 10	114
Three Queries of Type 11	116
Three Queries of Type 12	117
Three Queries of Type 13	119
One Query of Type 14.....	121
One Query of Type 15.....	121
One Query for Type 16:	123

One Query of Type 17:.....	125
Task 7. Web Database Application and its execution.....	125
7.1 Web Database application source program and screenshots showing its successful compilation	125
7.2 Screenshots showing the testing of the Web database application	129

Task 1

1.1 ER Diagram



1.2 Relational Schema

Customer (cname ,address ,category)

Assembly (assembly-id ,date_ordered ,assembly-details)

Process (process_id , process_data)

Order (assembly id ,cname)

Fit_Process (process_id , fit_type)

Paint_Process (process_id , paint_type, paint_method)

Cut_Process (process_id , cutting_type , machine_type)

Job (job_id , date_commenced, date_completed, labor_time, job_type)

Cut-Job (job_id, machine_type , machine_time, material ,labor_time)

Paint_Job (job_id, color, volume, labor_time)

Fit_Job (job_id , labor_time)

Account (account_id , date_established)

Department account (account_id, sup_cost)

Assembly account (account_id, sup_cost)

Process account (account_id, sup_cost)

Transaction (transaction_id, sup_cost)

Department (dept_id, dept_data)

Assign(assembly_id, process_id, job_id)

Pass_Through(assembly_id, process_id)

Record(transaction_id, job_id)

Record_Process_Cost(account_id, process_id)

Record_Department_Cost(account_id, dept_id)

Record_Assembly_Cost(account_id, assembly_id)

Supervised_By (process_id, dept_id)

Update (transaction_id, account_id)

Task 2. Data Dictionary

Customer

Field Name	Data Type	Size	Constraints
<u>cname</u>	varchar	20 bytes	Primary Key
address	varchar	20 bytes	
category	Int	5bytes	1-10

Assemblies

Field Name	Data Type	Size	Constraints
<u>assembly_id</u>	int	5bytes	Primary Key
Date_ordered	DateTime	8 bytes	
Assembly_detains	varchar	20 bytes	

Assign

Field Name	Data Type	Size	Constraints
<u>Assembly_id</u>	int	5 bytes	Foreign Key (Assembly)

Process_id	int	5 bytes	Foreign Key (Process)
Job_id	Int	5 bytes	Primary Key, Foreign Key (Job)

Pass_Through

Field Name	Data Type	Size	Constraints
Assembly_id	int	5 bytes	Foreign Key (Assembly)
Process_id	int	5 bytes	Primary Key, Foreign Key (Process)

Process

Field Name	Data Type	Size	Constraints
process_id	int	5 bytes	Primary Key
Process_data	varchar	20 bytes	

Order

Field Name	Data Type	Size	Constraints
Assembly_id	int	5 bytes	Primary Key, Foreign Key (Assembly)
cname	int	5 bytes	Foreign Key (Customer)

Fit_Process

Field Name	Data Type	Size	Constraints
Process_id	int	5 bytes	Primary Key, Foreign Key (Process)
Fit_type	varchar	20 bytes	

Paint_Process

Field Name	Data Type	Size	Constraints
process_id	int	5 bytes	Primary Key, Foreign Key (Process)
Paint_type	varchar	10 bytes	
Paint_method	varchar	10 bytes	

Cut_process

Field Name	Data Type	Size	Constraints
process_id	int	5 bytes	Primary Key, Foreign Key (Process)

Machine_type	varchar	10 bytes	
Cut_type	varchar	10 bytes	

Job

Field Name	Data Type	Size	Constraints
Job_id	int	5 bytes	Primary Key
Date_commenced	Date	3 bytes	
Date_completed	Date	3 bytes	
Labor_time	int	5 bytes	
Job_type	int	5 bytes	1-3

Cut_Job

Field Name	Data Type	Size	Constraints
job_id	int	5 bytes	Primary Key, Foreign Key (Job)
Machine_type	varchar	20 bytes	
Machine_time	int	5 bytes	
Material	varchar	20 bytes	
Labor_time	int	5 bytes	

Paint_Job

Field Name	Data Type	Size	Constraints
job_id	int	5 bytes	Primary Key, Foreign Key (Job)
color	varchar	10 bytes	
volume	int	5 bytes	
Labor_time	int	5 bytes	

Fit_Job

Field Name	Data Type	Size	Constraints
job_id	int	5 bytes	Primary Key, Foreign Key (Job)
Labor_time	int	5 bytes	

Account

Field Name	Data Type	Size	Constraints
Account_id	int	5 bytes	Primary Key
Date_established	Date	3 bytes	

Department

Field Name	Data Type	Size	Constraints
Department_id	int	5 bytes	Primary Key
Date_established	Date	3 bytes	

Dept_Account

Field Name	Data Type	Size	Constraints
Account_id	int	5 bytes	Primary Key, Foreign key (Account)
Sup_cost	real	4 bytes	

Assembly_Account

Field Name	Data Type	Size	Constraints
Account_id	int	5 bytes	Primary Key, Foreign key (Account)
Sup_cost	real	4 bytes	

Process_Account

Field Name	Data Type	Size	Constraints
Account_id	int	5 bytes	Primary Key, Foreign key (Account)
Sup_cost	real	4 bytes	

Transactions

Field Name	Data Type	Size	Constraints
Account_id	int	5 bytes	Primary Key, Foreign key (Account)
Sup_cost	real	4 bytes	

Record

Field Name	Data Type	Size	Constraints
Transaction_id	int	5 bytes	Primary Key, Foreign key (Transactions)
Job_id	int	5 bytes	Foreign Key (Job)

Record_Dept_Cost

Field Name	Data Type	Size	Constraints

Account_id	int	5 bytes	Primary Key, Foreign key (Account)
Dept_id	int	5 bytes	Foreign Key (Department)

Record_Process_Cost

Field Name	Data Type	Size	Constraints
account_id	int	5 bytes	Primary Key, Foreign key (account)
process_id	int	5 bytes	Foreign Key (Process)

Record_Assembly_Cost

Field Name	Data Type	Size	Constraints
accound_id	int	5 bytes	Primary Key, Foreign key (Account)
Assembly_id	int	5 bytes	Foreign Key (Job)

Supervised_By

Field Name	Data Type	Size	Constraints
Process_id	int	5 bytes	Primary Key, Foreign key (Process)
Dept_id	int	5 bytes	Foreign Key (Department)

Update

Field Name	Data Type	Size	Constraints
Transaction_id	int	5 bytes	Primary Key, Foreign key (Transactions)
Account_id	int	5 bytes	Foreign Key (Account)

Task 3

3.1 Discussion of Storage Structures for Tables

Table Name	Query # and Type	Search Key	Query Frequency	Selected File Organization	Justifications
Customer	1.) Insert	Category	30/Day 100/Day	Index Sequential on	For range Search we

	13.) Range Search			Search key Category	need Index sequential file organization.
Assemblies	4.) Insert	NA	40/Day	Heap File Organization	Heap file organization is great for insertions
Assign	6.) insert 14.) Range Search	Job_id	50/Day 1/Month	Heap File Organization	Heap file organization is great for insertions. Range Search is infrequent.
Process	3.) Insert	NA	Infrequent	Heap File Organization	Heap file organization is great for insertions
Department	3.) Insert	NA	Infrequent	Heap File Organization	Heap file organization is great for insertions
Orders	4.) Insert	NA	40/Day	Heap File Organization	Heap file organization is great for insertions
Pass_Through	4.) Insert 11.) Random Search	Process_id	40/Day 100/Day	Extendable hashing with key 'process_id'	Extendable hashing is the best for random searches.
Account	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'process_id'	Extendable hashing is the best for random searches.
Record	8.) Insert	NA	50/Day	Heap File Organization	Heap file organization is great for insertions
Record_Dept_Cost	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'process_id'	Extendable hashing is the best for random searches.
Record_Assembly_Cost	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with	Extendable hashing is the best for

				key 'process_id'	random searches.
Record_Process_Cost	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'process_id'	Extendable hashing is the best for random searches.
Job	6.) Insert 7.) Random Search 10.) Random Search 14.) Range Search	Job_id Job_id	50/Day 50/Day 20/Day	Extendable hashing with key 'process_id'	Extendable hashing is the best for random searches. Range Search is infrequent.
Cut_Job	7.) Insert 14.) Range Search	Job_id	50/Day 1/Month	Heap File Organization	Heap file organization is great for insertions. Range search is infrequent.
Fit_Job	7.) Insert	NA	50/Day	Heap File Organization	Heap file organization is great for insertions.
Paint_job	7.) Insert 15.) Random Search	Job_id	50/Day 1/Week	Heap File Organization	Heap file organization is great for insertions. Random Search is infrequent by comparison.
Transactions	8.) Insert	NA	50/Day	Heap File Organization	Heap file organization is great for insertions.
Updates	8.) Insert	NA	50/Day	Heap File Organization	Heap file organization is great for insertions.
Dept_Account	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'account_id'	Extendable hashing is the best for random searches.

Process_Account	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'account_id'	Extendable hashing is the best for random searches.
Assembly_Account	5.) Insert 8.) Random Search	Account_id	10/Day 50/Day	Extendable hashing with key 'account_id'	Extendable hashing is the best for random searches.
Cut_Process	3.) insert	NA	Infrequent	Heap File Organization	Heap file organization is great for insertions.
Fit_Process	3.) insert	NA	Infrequent	Heap File Organization	Heap file organization is great for insertions.
Paint_Process	3.) insert	NA	Infrequent	Heap File Organization	Heap file organization is great for insertions.
Supervised_By	3.) Insert 8.) Random Search 10.) Random Search 11.) Random Search 12.) Random Search	Process_id Dept_id Process_id Dept_id	Infrequent 50/Day 20/Day 100/Day 20/Day	Extendable hashing file organization with key 'process_id'	Extendable hashing is the best for random searches.

3.2 Discussion of storage structures for tables (Azure SQL Database)

I could not determine a way to create different storage structures for the Tables on Azure SQL Database.

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

SQLQuery_3.sql - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ... Welcome SQLQuery_3.sql - gaur0001...ur0001) X
home > ashesh > Documents > DBMS > Final_Project > SQLQuery_3.sql | Explain Enable SQLCMD
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain Enable SQLCMD
Tables
1 DROP TABLE IF EXISTS Orders;
2 DROP TABLE IF EXISTS Pass_Through;
3 DROP TABLE IF EXISTS Record_Assembly_Cost;
4 DROP TABLE IF EXISTS Record_Assembly_Cost_By;
5 DROP TABLE IF EXISTS Customer;
6 DROP TABLE IF EXISTS Record;
7 DROP TABLE IF EXISTS Fit_Job;
8 DROP TABLE IF EXISTS Cut_Job;
9 DROP TABLE IF EXISTS Cut_Job;
10 DROP TABLE IF EXISTS Job;
11 DROP TABLE IF EXISTS Fit_Process;
12 DROP TABLE IF EXISTS Cut_Process;
13 DROP TABLE IF EXISTS Record_Assembly_Cost;
14 DROP TABLE IF EXISTS Record_Assembly_Cost;
15 DROP TABLE IF EXISTS Record_Dept_Cost;
16 DROP TABLE IF EXISTS Record_Process_Cost;
17 DROP TABLE IF EXISTS Dept_Account;
18 DROP TABLE IF EXISTS Record_Assembly_Cost;
19 DROP TABLE IF EXISTS Process_Accounts;
20 DROP TABLE IF EXISTS Updates;
21 DROP TABLE IF EXISTS Account;
22 DROP TABLE IF EXISTS Transactions;
23 DROP TABLE IF EXISTS Department;
24 DROP TABLE IF EXISTS Record_Assembly_Cost;
25 DROP TABLE IF EXISTS Record_Assembly_Cost;
26
CREATE TABLE Customer(
    name VARCHAR(20),
    address VARCHAR(20),
    category INT,
    PRIMARY KEY(name),
    CONSTRAINT chk_category CHECK (category IN ('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'))
);
CREATE TABLE Department(
    dept_id INT,
    dept_data VARCHAR(10),
    PRIMARY KEY (dept_id)
);
CREATE TABLE Assemblies(
    assembly_id INT PRIMARY KEY,
    assembly_details VARCHAR(64),
    date_ordered INT,
);
CREATE TABLE Process(
    process_id INT,
    process_data VARCHAR(20),
    PRIMARY KEY (process_id)
);
CREATE TABLE Job(
    job_id INT,
    date_commenced INT,
    date_completed INT,
    label_type INT,
    job_type INT,
    PRIMARY KEY (job_id),
    CONSTRAINT chk_job_type CHECK (job_type IN ('1', '2', '3'))
);
CREATE TABLE Fit_Process(
    process_id INT PRIMARY KEY REFERENCES Process(process_id),
    fit_type VARCHAR(20),
);
Messages
11:22:13 AM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:01.875

```

In 19, Col 38 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:01 gaur0001-sql-server.database.windows.net:cs-dsa-4513-sql-db

SQLQuery_3.sql - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ... Welcome SQLQuery_3.sql - gaur0001...ur0001) X
home > ashesh > Documents > DBMS > Final_Project > SQLQuery_3.sql | Explain Enable SQLCMD
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain Enable SQLCMD
Tables
1 DROP TABLE IF EXISTS Orders;
2 DROP TABLE IF EXISTS Pass_Through;
3 DROP TABLE IF EXISTS Record_Assembly_Cost;
4 DROP TABLE IF EXISTS Record_Assembly_Cost_By;
5 DROP TABLE IF EXISTS Customer;
6 DROP TABLE IF EXISTS Record;
7 DROP TABLE IF EXISTS Fit_Job;
8 DROP TABLE IF EXISTS Cut_Job;
9 DROP TABLE IF EXISTS Cut_Job;
10 DROP TABLE IF EXISTS Job;
11 DROP TABLE IF EXISTS Fit_Process;
12 DROP TABLE IF EXISTS Cut_Process;
13 DROP TABLE IF EXISTS Record_Assembly_Cost;
14 DROP TABLE IF EXISTS Record_Assembly_Cost;
15 DROP TABLE IF EXISTS Record_Dept_Cost;
16 DROP TABLE IF EXISTS Record_Process_Cost;
17 DROP TABLE IF EXISTS Dept_Account;
18 DROP TABLE IF EXISTS Record_Assembly_Cost;
19 DROP TABLE IF EXISTS Process_Accounts;
20 DROP TABLE IF EXISTS Updates;
21 DROP TABLE IF EXISTS Account;
22 DROP TABLE IF EXISTS Transactions;
23 DROP TABLE IF EXISTS Department;
24 DROP TABLE IF EXISTS Record_Assembly_Cost;
25 DROP TABLE IF EXISTS Record_Assembly_Cost;
26
CREATE TABLE Customer(
    name VARCHAR(20),
    address VARCHAR(20),
    category INT,
    PRIMARY KEY(name),
    CONSTRAINT chk_category CHECK (category IN ('1', '2', '3', '4', '5', '6', '7', '8', '9', '10'))
);
CREATE TABLE Department(
    dept_id INT,
    dept_data VARCHAR(10),
    PRIMARY KEY (dept_id)
);
CREATE TABLE Assemblies(
    assembly_id INT PRIMARY KEY,
    assembly_details VARCHAR(64),
    date_ordered INT,
);
CREATE TABLE Process(
    process_id INT,
    process_data VARCHAR(20),
    PRIMARY KEY (process_id)
);
CREATE TABLE Job(
    job_id INT,
    date_commenced INT,
    date_completed INT,
    label_type INT,
    job_type INT,
    PRIMARY KEY (job_id),
    CONSTRAINT chk_job_type CHECK (job_type IN ('1', '2', '3'))
);
CREATE TABLE Fit_Process(
    process_id INT PRIMARY KEY REFERENCES Process(process_id),
    fit_type VARCHAR(20),
);
Messages
11:22:13 AM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:01.875

```

In 19, Col 38 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:01 gaur0001-sql-server.database.windows.net:cs-dsa-4513-sql-db

SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ... Welcome SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio
home > ashesh > Documents > DBMS > Final_Project > SQLQuery_3.sql
Run Cancel ⚡ Disconnect ⚡ Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

64 CREATE TABLE Fit_Process(
65     process_id INT PRIMARY KEY REFERENCES Process(process_id),
66     fit_type VARCHAR(20),
67 );
68
69 );
70
71 CREATE TABLE Paint_Process(
72     process_id INT PRIMARY KEY REFERENCES Process(process_id),
73     paint_type VARCHAR(20),
74     painting_method VARCHAR(20),
75 );
76
77 CREATE TABLE Cut_Process(
78     process_id INT PRIMARY KEY REFERENCES Process(process_id),
79     cutting_type VARCHAR(20),
80     machine_type VARCHAR(20),
81 );
82
83 );
84
85 CREATE TABLE Orders(
86     cname VARCHAR(20),
87     assembly_id INT,
88     PRIMARY KEY (cname, assembly_id),
89     FOREIGN KEY (cname) REFERENCES Customer(cname),
90     FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id)
91 );
92
93 CREATE TABLE Pass_Through(
94     assembly_id INT,
95     process_id INT,
96     PRIMARY KEY (assembly_id, process_id),
97     FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id),
98     FOREIGN KEY (process_id) REFERENCES Process(process_id),
99 );
100
101 CREATE TABLE Cut_Job(
102     job_id INT PRIMARY KEY REFERENCES Job(job_id),
103     machine_type VARCHAR(20),
104     machine_time INT,
105     material_used VARCHAR(20),
106 );
107
108
109 CREATE TABLE Paint_Job(
110     job_id INT PRIMARY KEY REFERENCES Job(job_id),
111     color VARCHAR(10),
112     volume INT,
113     labor_time INT,
114 );
115
116 CREATE TABLE Fit_Job(
117     job_id INT PRIMARY KEY REFERENCES Job(job_id),
118     labor_time INT,
119 );
120
121 CREATE TABLE Transactions(
122     transaction_id INT PRIMARY KEY,
123     sup_cost REAL
124 );
125
126 CREATE TABLE Account(
127     account_id INT PRIMARY KEY,
128     account_date INT
129 );
130
131 CREATE TABLE Dept_Account(
132     account_id INT PRIMARY KEY REFERENCES Account(account_id),
133     sup_cost REAL
134 );
135
136 CREATE TABLE Assembly_Account(
137     account_id INT PRIMARY KEY REFERENCES Account(account_id),
138     sup_cost REAL
139 );
140
141 CREATE TABLE Process_Account(
142     account_id INT PRIMARY KEY REFERENCES Account(account_id),
143
Messages
11:22:13 AM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:01.875
Ln 19, Col 38 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:01 gaur001-sql-server.database.windows.net: cs-dsa-4513-sql-db ⚡

```

SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ... Welcome SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio
home > ashesh > Documents > DBMS > Final_Project > SQLQuery_3.sql
Run Cancel ⚡ Disconnect ⚡ Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

101 Create TABLE Cut_Job(
102     job_id INT PRIMARY KEY REFERENCES Job(job_id),
103     machine_type VARCHAR(20),
104     machine_time INT,
105     material_used VARCHAR(20),
106     labor_time INT
107 );
108
109 CREATE TABLE Paint_Job(
110     job_id INT PRIMARY KEY REFERENCES Job(job_id),
111     color VARCHAR(10),
112     volume INT,
113     labor_time INT,
114 );
115
116 CREATE TABLE Fit_Job(
117     job_id INT PRIMARY KEY REFERENCES Job(job_id),
118     labor_time INT,
119 );
120
121 CREATE TABLE Transactions(
122     transaction_id INT PRIMARY KEY,
123     sup_cost REAL
124 );
125
126 CREATE TABLE Account(
127     account_id INT PRIMARY KEY,
128     account_date INT
129 );
130
131 CREATE TABLE Dept_Account(
132     account_id INT PRIMARY KEY REFERENCES Account(account_id),
133     sup_cost REAL
134 );
135
136 CREATE TABLE Assembly_Account(
137     account_id INT PRIMARY KEY REFERENCES Account(account_id),
138     sup_cost REAL
139 );
140
141 CREATE TABLE Process_Account(
142     account_id INT PRIMARY KEY REFERENCES Account(account_id),
143
Messages
11:22:13 AM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:01.875
Ln 19, Col 38 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:01 gaur001-sql-server.database.windows.net: cs-dsa-4513-sql-db ⚡

```

SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ...
SERVERS gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001)
Tables ...
    > dbo.Account
    > dbo.Acted
    > dbo.Assembly
    > dbo.Assembly_Account
    > dbo.Assign
    > dbo.Customer
    > dbo.Cut_Job
    > dbo.Cut_Process
    > Columns ...
    > Keys ...
        → PK_Fit_Proc_9446C3E1153...
        → FK_Fit_Proce_proce_000A...
    > Constraints ...
    > Triggers ...
    > Indexes ...
    > Statistics ...
    > dbo.job
    > dbo.Movie
    > dbo.Orders
    > dbo.Paint_Job
    > dbo.Paint_Process
    > dbo.Pass_Through
    > dbo.Perform
    > dbo.Process
    > dbo.Process_Account
    > dbo.Record
    > dbo.Record_Assembly_Cost
    > dbo.Record_Dept_Cost
    > dbo.Record_Process_Cost
    > dbo.Supervised_By
    > ...
    > Transmissions ...
AZURE ...
SQL SERVER BIG DATA CLUSTERS
125 △ 0

```

Welcome SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) ×

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

```

139 );
140
141 CREATE TABLE Process_Account(
142     account_id INT PRIMARY KEY REFERENCES Account(account_id),
143     sup_cost REAL
144 );
145
146 CREATE TABLE Record(
147     transaction_id INT,
148     job_id INT,
149     PRIMARY KEY(transaction_id, job_id),
150     FOREIGN key(transaction_id) REFERENCES Transactions(transaction_id),
151     FOREIGN key(job_id) REFERENCES Job(job_id)
152 );
153
154
155 CREATE TABLE Asigned(
156     job_id INT,
157     assembly_id INT,
158     process_id INT,
159     FOREIGN key(job_id) REFERENCES Job(job_id),
160     FOREIGN key(assembly_id) REFERENCES Assemblies(assembly_id),
161     FOREIGN key(process_id) REFERENCES Process(process_id)
162 );
163
164
165 CREATE TABLE Updates(
166     transaction_id INT,
167     account_id INT,
168     PRIMARY KEY (transaction_id, account_id),
169     FOREIGN key(transaction_id) REFERENCES Transactions(transaction_id),
170     FOREIGN key(account_id) REFERENCES Account(account_id)
171 );
172
173
174 CREATE TABLE Supervised_By(
175     process_id INT,
176     dept_id INT,
177     PRIMARY KEY (process_id, dept_id),
178     FOREIGN key(process_id) REFERENCES Process(process_id),
179     FOREIGN key(dept_id) REFERENCES Department(dept_id)
180 );
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201 );

```

Messages

11:22:13 AM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:01.875

Ln 19, Col 38 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:01 gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db AP D

SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio

```

File Edit View Help
CONNECTIONS ...
SERVERS gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001)
Tables ...
    > dbo.Account
    > dbo.Acted
    > dbo.Assembly
    > dbo.Assembly_Account
    > dbo.Assign
    > dbo.Customer
    > dbo.Cut_Job
    > dbo.Cut_Process
    > dbo.Department
    > dbo.Dept_Account
    > dbo.Director
    > dbo.Fit_Job
    > dbo.Fit_Process
    > Columns ...
    > Keys ...
        → PK_Fit_Proc_9446C3E1153...
        → FK_Fit_Proce_proce_000A...
    > Constraints ...
    > Triggers ...
    > Indexes ...
    > Statistics ...
    > dbo.job
    > dbo.Movie
    > dbo.Orders
    > dbo.Paint_Job
    > dbo.Paint_Process
    > dbo.Pass_Through
    > dbo.Perform
    > dbo.Process
    > dbo.Process_Account
    > dbo.Record
    > dbo.Record_Assembly_Cost
    > dbo.Record_Dept_Cost
    > dbo.Record_Process_Cost
    > dbo.Supervised_By
    > ...
    > Transmissions ...
AZURE ...
SQL SERVER BIG DATA CLUSTERS
125 △ 0

```

Welcome SQLQuery_3.sql - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) ×

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

```

173
174 CREATE TABLE Supervised_By(
175     process_id INT,
176     dept_id INT,
177     PRIMARY KEY (process_id, dept_id),
178     FOREIGN key(process_id) REFERENCES Process(process_id),
179     FOREIGN key(dept_id) REFERENCES Department(dept_id)
180 );
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201 );

```

Messages

12:05:28 PM Started executing query at Line 1
Commands completed successfully.
Total execution time: 00:00:00.886

Ln 181, Col 31 Spaces: 4 UTF-8 LF SQL MSSQL 0 rows 00:00:05 gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db AP D

Task 5

5.1 SQL statements (and Transact SQL stored procedures, if any) Implementing all queries (1-15 and error checking)

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer HelloWorld.java AzureSQLJDBC.java
src
  AzureSql
    src
      (default package)
        AzureSQLJDBC.java
  JRE System Library [JavaSE-12]
  Referenced Libraries
HelloWorld

1* import java.sql.Connection;
2
3  public class AzureSQLJDBC {
4
5    // Database credentials
6    final static String HOSTNAME = "gaur0001sql-server.database.windows.net";
7    final static String DBNAME = "tcs-dsa-4513-sql-db";
8    final static String USERNAME = "gaur0001";
9    final static String PASSWORD = "ouNetPass1!";
10
11   // Database connection string
12   final static String URL = String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=%s.database.windows.net", HOSTNAME, DBNAME, USERNAME, PASSWORD);
13
14   // Query templates
15   final static String QUERY_TEMPLATE_1_a = "delete From Cut_Job Where job_id > (?) and job_id < (?) ";
16   final static String QUERY_TEMPLATE_1_b = "Delete From Assign Where job_id > (?) and job_id < (?) ";
17   final static String QUERY_TEMPLATE_1_c = "Delete From Job Where job_id > (?) and job_id < (?) ";
18   final static String QUERY_TEMPLATE_15 ="UPDATE Paint_Job "
19   "SET color = (?)"
20   "Where job_id = (?);"
21
22   final static String QUERY_TEMPLATE_2_a = "UPDATE Job "
23   "SET date_completed = (?), labor_time = (?), job_type = (?) "
24   "Where job_id = (?);"
25
26   final static String QUERY_TEMPLATE_7_d ="INSERT into Cut_Job "
27   "VALUES (? ,? ,? ,? );"
28
29   final static String QUERY_TEMPLATE_7_c ="INSERT into Paint_Job "
30   "VALUES (? ,? ,? ,? );"
31
32   final static String QUERY_TEMPLATE_7_b ="INSERT into Fit_Job "
33   "VALUES (? ,? );"
34
35   final static String QUERY_TEMPLATE_10_a = "INSERT INTO Customer "
36   "VALUES (? ,? ,? );"
37
38   final static String QUERY_TEMPLATE_10_b ="SELECT DISTINCT process_id "
39   "FROM Supervised_By WHERE dept_id = (?);"
40
41   final static String QUERY_TEMPLATE_10_c ="SELECT distinct job_id "
42   "FROM Assign WHERE process_id = (?);"
43
44   final static String QUERY_TEMPLATE_10_d = "SELECT labor_time, date_completed "
45   "FROM Job WHERE job_id = (?);"
46
47   final static String QUERY_TEMPLATE_2_a = "INSERT INTO Department "
48   "VALUES (? ,? );"
49
50   final static String QUERY_TEMPLATE_3_a = "INSERT INTO Process "
51   "VALUES (? ,? );"
52
53   final static String QUERY_TEMPLATE_3_b = "INSERT INTO Fit_Process "
54   "VALUES (? ,? );"
55
56   final static String QUERY_TEMPLATE_3_c = "INSERT INTO Cut_Process "
57   "VALUES (? ,? );"
58
59   final static String QUERY_TEMPLATE_3_d = "INSERT into Supervised_By "
60   "VALUES (? ,? ,? );"
61
62   final static String QUERY_TEMPLATE_8_a = "INSERT into Transactions "
63   "VALUES (? ,? );"
64
65   final static String QUERY_TEMPLATE_8_b = "INSERT into Record "
66   "VALUES (? ,? );"
67
68   final static String QUERY_TEMPLATE_8_c = "SELECT assembly_id, process_id from Assign "
69   "WHERE job_id = (?);"
70
71   final static String QUERY_TEMPLATE_8_d = "SELECT dept_id FROM Supervised_By "
72   "WHERE process_id =(?);"
73
74   final static String QUERY_TEMPLATE_8_e = "SELECT account_id FROM Record_Dept_Cost "
75   "WHERE assembly_id = (?);"
76
77   final static String QUERY_TEMPLATE_8_f = "SELECT account_id FROM Record_Assembly_Cost "
78   "WHERE assembly_id = (?);"
79
80   final static String QUERY_TEMPLATE_8_g = "SELECT account_id FROM Record_Process_Cost "
81   "WHERE process_id = (?);"
82
83   final static String QUERY_TEMPLATE_8_h = "SELECT sup_cost FROM Dept_Account "
84   "WHERE account_id = (?);"
85
86   final static String QUERY_TEMPLATE_8_i = "SELECT sup_cost FROM Assembly_Account "
87   "WHERE account_id =(?);"
88
89   final static String QUERY_TEMPLATE_8_j = "SELECT sup_cost FROM Process_Account "
90   "WHERE account_id = (?);"
91
92   final static String QUERY_TEMPLATE_8_k = "UPDATE Dept_Account "
93   "SET sup_cost = (?)"
94   "Where account_id = (?);"
95
96   final static String QUERY_TEMPLATE_8_l = "UPDATE Assembly_Account "
97   "SET sup_cost = (?)"
98
99   final static String QUERY_TEMPLATE_8_m = "UPDATE Process_Account "
100  "SET sup_cost = (?)"
101
102
103
104  final static String QUERY_TEMPLATE_11 = "select Supervised_By.process_id, dept_id "

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer HelloWorld.java AzureSQLJDBC.java
src
  AzureSql
    src
      (default package)
        AzureSQLJDBC.java
  JRE System Library [JavaSE-12]
  Referenced Libraries
HelloWorld

54
55   final static String QUERY_TEMPLATE_3 = "INSERT INTO Process "
56   "VALUES (? ,? );"
57
58   final static String QUERY_TEMPLATE_3_a = "INSERT INTO Fit_Process "
59   "VALUES (? ,? );"
60
61   final static String QUERY_TEMPLATE_3_b = "INSERT INTO Paint_Process "
62   "VALUES (? ,? );"
63
64   final static String QUERY_TEMPLATE_3_c = "INSERT INTO Cut_Process "
65   "VALUES (? ,? );"
66
67   final static String QUERY_TEMPLATE_3_d = "INSERT into Supervised_By "
68   "VALUES (? ,? ,? );"
69
70
71   final static String QUERY_TEMPLATE_8_a = "INSERT into Transactions "
72   "VALUES (? ,? );"
73
74   final static String QUERY_TEMPLATE_8_b = "INSERT into Record "
75   "VALUES (? ,? );"
76
77   final static String QUERY_TEMPLATE_8_c = "SELECT assembly_id, process_id from Assign "
78   "WHERE job_id = (?);"
79
80   final static String QUERY_TEMPLATE_8_d = "SELECT dept_id FROM Supervised_By "
81   "WHERE process_id =(?);"
82
83   final static String QUERY_TEMPLATE_8_e = "SELECT account_id FROM Record_Dept_Cost "
84   "WHERE assembly_id = (?);"
85
86   final static String QUERY_TEMPLATE_8_f = "SELECT account_id FROM Record_Assembly_Cost "
87   "WHERE assembly_id = (?);"
88
89   final static String QUERY_TEMPLATE_8_g = "SELECT account_id FROM Record_Process_Cost "
90   "WHERE process_id = (?);"
91
92   final static String QUERY_TEMPLATE_8_h = "SELECT sup_cost FROM Dept_Account "
93   "WHERE account_id = (?);"
94
95   final static String QUERY_TEMPLATE_8_i = "SELECT sup_cost FROM Assembly_Account "
96   "WHERE account_id =(?);"
97
98   final static String QUERY_TEMPLATE_8_j = "SELECT sup_cost FROM Process_Account "
99   "WHERE account_id = (?);"
100
101
102
103
104  final static String QUERY_TEMPLATE_11 = "select Supervised_By.process_id, dept_id "

```

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer [HelloWorld.java] [AzureSQLJDBC.java]
103
104    final static String QUERY_TEMPLATE_1 = "select Supervised_By.process_id, dept_id "
105        + "from Assemblies, Process, Supervised_By "
106        + "where Supervised_By.process_id = Process.process_id "
107        + "and Assembly.Assembly_id = (?) "
108        + "order by date_ordered ASC";
109
110    final static String QUERY_TEMPLATE_2 = "INSERT into Account " +
111        "VALUES (? ,?)";
112    final static String QUERY_TEMPLATE_3_b = "INSERT into Dept_Account " +
113        "VALUES (? ,?)";
114    final static String QUERY_TEMPLATE_5_c = "INSERT into Assembly_Account " +
115        "VALUES (? ,?)";
116    final static String QUERY_TEMPLATE_3_d = "INSERT into Process_Account " +
117        "VALUES (? ,?)";
118    final static String QUERY_TEMPLATE_3_e = "INSERT into Record_Dept_Cost " +
119        "VALUES (? ,?)";
120    final static String QUERY_TEMPLATE_5_f = "INSERT into Record_Assembly_Cost " +
121        "VALUES (? ,?)";
122    final static String QUERY_TEMPLATE_3_g = "INSERT into Record_Process_Cost " +
123        "VALUES (? ,?)";
124
125    final static String QUERY_TEMPLATE_12_a = "select process_id from Supervised_By " +
126        "where dept_id = (?)";
127    final static String QUERY_TEMPLATE_12_b = "select job_id, assembly_id from Assign " +
128        "where process_id = (?)";
129    final static String QUERY_TEMPLATE_12_c = "select job_type from Job " +
130        "where job_id = (?) and date_completed = (?)";
131
132    final static String QUERY_TEMPLATE_6_a = "INSERT into Job " +
133        "VALUES (? ,? ,? ,? )";
134    final static String QUERY_TEMPLATE_6_b = "INSERT into Assign " +
135        "VALUES (? ,? ,? )";
136
137    final static String QUERY_TEMPLATE_13 = "SELECT * FROM Customer WHERE category >= (?) and category < (?) " ;
138
139
// User input prompt/
140    final static String PROMPT =
141        "Please select one of the options below: \n" +
142        "1) Enter new customer; \n" +
143        "2) Enter a new department; \n" +
144        "3) Enter a new Process ; \n" +
145        "4) Enter a new Assembly; \n" +
146        "5) Enter a new Account Number;\n" +
147        "6) Enter a new Job;\n" +
148        "7) Enter the Completion date of a Job; \n" +
149        "8) Enter a new Transaction Number; \n" +
150        "9) Retrieve the total cost incurred on an assembly-id \n" +
151        "10)Retrieve total labor time within a Department for a given day: \n" +
152        "11)Retrieve the process through which a given assembly-id has passed: \n" +
153        "12)Retrieves the jobs completed during given date in a given department;

```

The errors:

Primary Key Constraint error check by Azure SQL Database

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer [HelloWorld.java] [AzureSQLJDBC.java]
124
125    final static String QUERY_TEMPLATE_12_a = "select process_id from Supervised_By " +
126        "where dept_id = (?)";
127    final static String QUERY_TEMPLATE_12_b = "select job_id, assembly_id from Assign " +
128        "where process_id = (?)";
129
130
Problems [avadoc Declaration Console]
<terminated> AzureSQLJDBC [Java Application] /usr/lib/jvm/java-16-openjdk/bin/java (Nov 22, 2021, 12:20:10 PM - 12:21:32 PM)
Done. 1 rows inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Insert or update New customers from a data file;
17) Export Customer Data
20) Exit!
1
Please enter customer cname:
ashesh
Please enter customer address:
Normal
Please enter customer category between 1 and 10:
2
Connecting to the database...
ashesh
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Customer'. Cannot insert duplicate key in object 'dbo.Customer'.
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1662)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:615)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:537)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3488)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:262)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeUpdate(SQLServerStatement.java:237)
at com.microsoft.sqlserver.jdbc@9.4.0.jre16/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:483)
at AzureSQLJDBC.main(AzureSQLJDBC.java:540)

```

Foreign Key Constraint error check by Azure SQL Database:

The screenshot shows the Eclipse IDE interface with the following details:

- File Structure:** Package Explorer shows a project named "AzureSql" with a "src" folder containing "HelloWorld.java" and "AzureSQLJDBC.java".
- Code Editor:** The "HelloWorld.java" file contains a main method with several options for interacting with the database. The user has selected option 5, which inserts data into the "account" table.
- Console Output:** The output window shows the execution of the Java application. It connects to the database and inserts data into the "account" table. However, it fails at step 11 (inserting into "dept") because it violates a FOREIGN KEY constraint named "FK_Record_De_dept__02B25B50".
- Stack Trace:** The stack trace details the exception thrown by Microsoft's JDBC driver for SQL Server, indicating the conflict occurred in the INSERT statement.

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer  HelloWorld.java  AzureSQLJDBC.java

Problems Javadoc Declaration Console

<terminated> AzureSQLJDBC [Java Application] /usr/lib/jvm/java-16-openjdk/bin/java (Nov 22, 2021, 12:29:57 PM - 12:30:26 PM)
Welcome to the sample application!

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10)Retrieve the process through which a given assembly-id has passed;
11)Retrieve the jobs completed during given date in a given department;
12)Retrieve the customers (in name order) whose category is in a given range;
14)Delete all cut-jobs with a Job Number;
15)Change the color of a Paint Job;
16)Import: enter new customers from a data file;
17)Export Customer Data
20) Exit!
5
Please enter a new Account number
1
Please enter the date of account establishment:
11102021
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 for Process Acc:
1
Please enter the department number associated with the account:
21
Connecting to the database...
Dispatching the query (5a)...
Done. 1 rows inserted in "departmentaccount" table.
Dispatching the query (5b)...
Done. 1 rows inserted in "departmentaccount" table.
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the FOREIGN KEY constraint "FK_Record_De_dept__02B25B50". The conflict occurred in
at com.microsoft.sqlserver.jdbc.Error.makeFromDriverError(SQLServerException.java:265)
at com.microsoft.sqlserver.jdbc.Error.makeFromDriverError(SQLServerStatement.java:1662)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.prepareStatement(SQLServerPreparedStatement.java:615)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.prepareStatement(SQLServerPreparedStatement.java:537)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.execute(IOBuffer.java:7417)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeCommand(SQLServerConnection.java:3488)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeCommand(SQLServerConnection.java:262)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.createStatement(SQLServerStatement.java:262)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeUpdate(SQLServerStatement.java:262)
at AzureSQLJDBC.main(AzureSQLJDBC.java:200)

```

Invalid query check by Azure SQL Database

The screenshot shows the Eclipse IDE interface with the following details:

- File Structure:** Package Explorer shows a project named "AzureSql" with a "src" folder containing "HelloWorld.java" and "AzureSQLJDBC.java".
- Code Editor:** The "HelloWorld.java" file contains a main method with several options for interacting with the database. The user has selected option 2, which inserts data into the "dept" table.
- Console Output:** The output window shows the execution of the Java application. It connects to the database and attempts to insert data into the "dept" table. However, it fails because the value for parameter number 2 (dept_no) is not set.
- Stack Trace:** The stack trace details the exception thrown by Microsoft's JDBC driver for SQL Server, indicating the value is not set for the parameter number 2.

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer  HelloWorld.java  AzureSQLJDBC.java

Problems Javadoc Declaration Console

<terminated> AzureSQLJDBC [Java Application] /usr/lib/jvm/java-16-openjdk/bin/java (Nov 22, 2021, 12:33:14 PM - 12:33:30 PM)
Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10)Retrieve the process through which a given assembly-id has passed;
11)Retrieve the jobs completed during given date in a given department;
12)Retrieve the customers (in name order) whose category is in a given range;
14)Delete all cut-jobs with a Job Number;
15)Change the color of a Paint Job;
16)Import: enter new customers from a data file;
17)Export Customer Data
20) Exit!
2
Please enter a new Account number
2
Please enter the date of account establishment:
11102021
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 for Process Acc:
1
Please enter the department number associated with the account:
2
Connecting to the database...
Dispatching the query (5a)...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The value is not set for the parameter number 2.
at com.microsoft.sqlserver.jdbc.Error.makeFromDriverError(SQLServerException.java:237)
at com.microsoft.sqlserver.jdbc.Error.makeFromDriverError(SQLServerPreparedStatement.java:438)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.prepareStatement(SQLServerPreparedStatement.java:391)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.prepareStatement.buildPreparedStatement(SQLServerPreparedStatement.java:569)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.execute(IOBuffer.java:7417)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeCommand(SQLServerConnection.java:3488)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeCommand(SQLServerConnection.java:262)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.createStatement(SQLServerStatement.java:237)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeUpdate(SQLServerStatement.java:237)
at com.microsoft.sqlserver.jdbc.ConnectionImpl.executeUpdate(SQLServerPreparedStatement.java:483)
at AzureSQLJDBC.main(AzureSQLJDBC.java:200)

```

CHECK Constraint check by Azure SQL Database

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The project is named "HelloWorld" under the "src" folder of the "AzureSql" package.
- Code Editor:** The code editor displays a Java file named "HelloWorld.java". The code includes logic for interacting with an Azure SQL database using JDBC.
- Output Window:** The "Console" tab shows the application's output. It prompts the user for various inputs (e.g., customer name, address, category) and then attempts to execute an INSERT statement. An error message is displayed: "com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint 'chk_category'. The conflict occurred in database 'cs-dsa-4513-sq'." This indicates a CHECK constraint violation.
- Java Class:** The "AzureSQLDBC.java" class is also visible in the package structure.

5.2 The Java source program and screenshots showing its successful compilation

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The project is named "HelloWorld" under the "src" folder of the "AzureSql" package.
- Code Editor:** The code editor displays the "AzureSQLDBC.java" file. The code contains static strings for database connection details and query templates for various database operations (e.g., insert, update, delete).
- Java Class:** The "HelloWorld.java" class is also visible in the package structure.

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

src - AzureSql

1 HelloWorld.java

```
58 final static String QUERY_TEMPLATE_A_b = "INSERT INTO Point_Process " +  
59 "VALUES (?, ?, ?);"  
60 final static String QUERY_TEMPLATE_A_c = "INSERT INTO Cut_Process " +  
61 "VALUES (?, ?, ?);"  
62 final static String QUERY_TEMPLATE_A_d = "INSERT into Supervised_By " +  
63 "VALUES (?, ?, ?);"  
64 final static String QUERY_TEMPLATE_B_a = "INSERT into Transactions " +  
65 "VALUES (?, ?, ?);"  
66 final static String QUERY_TEMPLATE_B_b = "INSERT into Record " +  
67 "VALUES (?, ?, ?);"  
68 final static String QUERY_TEMPLATE_C_c = "SELECT assembly_id, process_id From Assign " +  
69 "WHERE job_id = ?;";  
70 final static String QUERY_TEMPLATE_D_d = "SELECT dept_id FROM Supervised_By " +  
71 "WHERE process_id =?;";  
72 final static String QUERY_TEMPLATE_E_e = "SELECT account_id FROM Record_Dept_Cost " +  
73 "WHERE dept_id = ?;";  
74 final static String QUERY_TEMPLATE_F_f = "SELECT account_id FROM Record_Assembly_Cost " +  
75 "WHERE process_id =?;";  
76 final static String QUERY_TEMPLATE_G_g = "SELECT account_id FROM Record_Process_Cost " +  
77 "WHERE process_id =?;";  
78 final static String QUERY_TEMPLATE_H_h = "SELECT sup_cost FROM Dept_Account " +  
79 "WHERE account_id =?;";  
80 final static String QUERY_TEMPLATE_I_i = "SELECT sup_cost FROM Dept_Account " +  
81 "WHERE account_id =?;";  
82 final static String QUERY_TEMPLATE_J_j = "SELECT sup_cost FROM Assembly_Account " +  
83 "WHERE account_id =?;";  
84 final static String QUERY_TEMPLATE_K_k = "SELECT sup_cost FROM Process_Account " +  
85 "WHERE account_id =?;";  
86 final static String QUERY_TEMPLATE_L_l = "UPDATE Dept_Account " +  
87 "SET sup_cost = ? WHERE account_id = ?;";  
88 final static String QUERY_TEMPLATE_M_m = "UPDATE Assembly_Account " +  
89 "SET sup_cost = ? WHERE account_id = ?;";  
90 final static String QUERY_TEMPLATE_N_n = "UPDATE Process_Account " +  
91 "SET sup_cost = ? WHERE account_id = ?;";  
92 final static String QUERY_TEMPLATE_O_o = "UPDATE Process " +  
93 "SET supervised_by_id = ? WHERE process_id = ?;";  
94 final static String QUERY_TEMPLATE_P_p = "UPDATE Assembly " +  
95 "SET process_id = ? WHERE assembly_id = ?;";  
96 final static String QUERY_TEMPLATE_Q_q = "INSERT into Assemblies " +  
97 "VALUES (?, ?, ?);"  
98 final static String QUERY_TEMPLATE_R_r = "INSERT into Orders " +  
99 "VALUES (?, ?, ?);"  
100 final static String QUERY_TEMPLATE_S_s = "INSERT into Pass_Through " +  
101 "VALUES (?, ?, ?, ?);"  
102  
103 final static String QUERY_TEMPLATE_T_t = "select Supervised_By.process_id, dept_id " +  
104 "from Assemblies_Process_Supervised_By " +  
105 "where Assemblies_Assembly_ID = ? and Process_Process_ID = " +  
106 "and Assemblies.Assembly_ID = ?"  
107 "order by date ordered ASC;"  
108  
109
```

```
eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
[HelloWorld] [HelloWorld.java]
1 package com.hello;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.Statement;
8
9 public class HelloWorld {
10     public static void main(String[] args) throws SQLException {
11         Connection connection = null;
12         Statement statement = null;
13         PreparedStatement preparedStatement = null;
14         ResultSet resultSet = null;
15
16         //11)Retrieve the customers (in home order) whose category is in a given range: <n> ~ <n>
17         //11)Delete all cut-jobs with a Job Number: <n>*
18         //12)Update the quantity of a specific item: <n>*
19         //16)Import enter new customers from a data file: <n>*
20         //17)Export customer data <n>*
21         //18)Exit[?]
22
23         public static void main(String[] args) throws SQLException {
24             System.out.println("Welcome to the sample application!");
25
26             final Scanner sc = new Scanner(System.in); // Scanner is used to collect the user input
27             String userOption = sc.nextLine(); // Collect user selection on pressing enter key
28             while (!userOption.equals("11")) { // If user enters 11 over for loop, then option 1 is selected
29                 System.out.print("Please enter the available options");
30                 option = sc.nextInt(); // Read in the user option selection
31
32                 switch (option) { // Switch between different options
33                     case 11:
34                         System.out.println("Please enter a new Account number:");
35                         final int accountNumber = sc.nextInt(); // Read in the user Input of student ID
36                         sc.nextLine();
37                         System.out.println("Please enter the date of account establishment:");
38                         final Date accountDate = sc.nextDate();
39
40                         System.out.println("Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 for Process Acc:");
41                         final int accType = sc.nextInt();
42                         sc.nextLine();
43                         if (accType == 1) {
44                             System.out.println("Please enter the department number associated with the account:");
45                             final int deptNum = sc.nextInt();
46                             sc.nextLine();
47                             System.out.println("Connecting to the database...");
```

eclipse-workspace - AzureSQL/src/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

W Package Explorer 1 HelloWorld.java (Resource History)

```
283
284     }
285
286     try {
287         final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_3_B) {
288             //Populate the query template with the data collected from the user
289             //System.out.println(name);
290             statement.setInt1(acnum);
291             statement.setString2(acname);
292
293             System.out.println("Dispatching the query 5b..."); //Actually execute the populated query
294             final int rows_inserted = statement.executeUpdate();
295             System.out.println(String.format("One %d rows inserted in \\\"departmentaccount\\\" table .", rows_inserted));
296         }
297     } catch (final SQLException e) {
298         e.printStackTrace();
299     }
300
301     try {
302         final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_5_B) {
303             //Populate the query template with the data collected from the user
304             //System.out.println(name);
305             statement.setInt1(acnum);
306             statement.setString2(acname);
307             statement.setInt2(deptNo);
308
309             System.out.println("Dispatching the query 5e..."); //Actually execute the populated query
310             final int rows_inserted = statement.executeUpdate();
311             System.out.println(String.format("One %d rows inserted in \\\"record_dept_cost\\\" table .", rows_inserted));
312         }
313     } catch (final SQLException e) {
314         e.printStackTrace();
315     }
316
317     else if (execType == 2) { //Get the assembly ID of the selected Process
318         System.out.println("Please enter the associated Assembly ID: ";
319         final String assemblyId = sc.nextLine();
320         sc.nextLine();
321
322         System.out.println("Connecting to the database..."); //Get a database connection and prepare a query statement
323         final Connection connection = DriverManager.getConnection(URL)); {
324
325             try {
326                 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_5_A) {
327                     //Populate the query template with the data collected from the user
328                     //System.out.println(name);
329                     statement1.setInt1(acnum);
330                     statement1.setString2(acname);
331
332                     System.out.println("Dispatching the query 5a..."); //Actually execute the populated query
333             }
```

eclipse-workspace - Azuresqj/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

src

Helloworld

+ AzureSql

+ (default package)

+ AzureSQLJDBC.java

+ IRE System Library [JavaSE-12]

+ Referenced Libraries

+ Helloworld

```
251     System.out.println("Dispathching the query 5(a)...");  
252     // Execute the query and get the populated statement  
253     final int rows_inserted = statement1.executeUpdate();  
254     System.out.println(String.format("Done. %d rows inserted in \\'Account\\' table.", rows_inserted));  
255 }  
try {  
    final PreparedStatement statements3 = connection.prepareStatement(QUERY_TEMPLATE_5_c) {  
        // Populate the query template with the data collected from the user  
        // (statements3.setNthParam(  
        statements3.setNthParam(1, accName);  
        statements3.setNthParam(2, accName);  
        statements3.setNthParam(3, 0);  
        System.out.println("Dispathching the query 5(c)...");  
        // Actually execute the populated query  
        final int rows_inserted = statements3.executeUpdate();  
        System.out.println(String.format("Done. %d rows inserted in \\'assemblyaccount\\' table.", rows_inserted));  
        System.out.println(String.format("Done. %d rows inserted in \\'assemblybyaccount\\' table.", rows_inserted));  
    }  
}  
try {  
    final PreparedStatement statements3 = connection.prepareStatement(QUERY_TEMPLATE_5_f) {  
        // Populate the query template with the data collected from the user  
        // (statements3.setNthParam(  
        statements3.setNthParam(1, accName);  
        statements3.setNthParam(2, accName);  
        statements3.setNthParam(3, 0);  
        System.out.println("Dispathching the query 5(f)...");  
        // Actually execute the populated query  
        final int rows_inserted = statements3.executeUpdate();  
        System.out.println(String.format("Executing Statement 5(f)..."));  
        System.out.println(String.format("Done. %d rows inserted in \\'record_assembly_cost\\' table.", rows_inserted));  
    }  
}  
}  
}  
else {  
    // Information for Cut Process  
    System.out.println("Please enter the associated process ID:");  
    final int processID = sc.nextInt();  
    sc.nextLine();  
    System.out.println("Connecting to the database...");  
    // Get a database connection and prepare a query statement  
    try (final Connection connection = DriverManager.getConnection(URL)) {  
        try {  
            final PreparedStatement statements1 = connection.prepareStatement(QUERY_TEMPLATE_5_a) {  
                // Populate the query template with the data collected from the user  
                // (statements1.setNthParam(  
                statements1.setNthParam(1, accName);  
                statements1.setNthParam(2, accName);  
                statements1.setNthParam(3, 0);  
                System.out.println("Dispathching the query 5(a)...");  
                final int rows_inserted = statements1.executeUpdate();  
                System.out.println(String.format("Done. %d rows inserted in \\'Account\\' table.", rows_inserted));  
            }  
        }
```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
src | eclipse-workspace | AzureSql | src | AzureSQLDBC.java | package-info.java |
+-- src
|   +-- com
|   |   +-- j2ee
|   |       +-- src
|   |           +-- com
|   |               +-- j2ee
|   |                   +-- src
|   |                       +-- AzureSQLDBC.java
|   +-- JRE System Library [javaSE-13]
|   +-- Referenced Libraries
+-- HelloWorld

```

```

1 package com.j2ee;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.Scanner;
9
10 public class AzureSQLDBC {
11
12     public static void main(String[] args) {
13         Connection connection = null;
14         PreparedStatement statement1 = null;
15         PreparedStatement statement2 = null;
16         PreparedStatement statement3 = null;
17         Scanner sc = new Scanner(System.in);
18
19         try {
20             // Establish the connection with the database
21             connection = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=master;user=sa;password=123456");
22
23             // Prepare the query template with the data collected from the user
24             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_3_a);
25             statement1.setString(1, "John");
26             statement1.setString(2, "Doe");
27             statement1.setInt(3, 20);
28
29             // Execute the populated query
30             final int rows_inserted = statement1.executeUpdate();
31             System.out.println(String.format("Done. %d row inserted in 'account' table.", rows_inserted));
32
33             // Dispatching the query 3(a)...*
34             System.out.println("Dispatching the query 3(a)...*");
35             final int rows_inserted = statement1.executeUpdate();
36             System.out.println(String.format("Done. %d row inserted in 'processaccount' table..", rows_inserted));
37
38             // Establish the connection with the database
39             connection = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=master;user=sa;password=123456");
40
41             // Prepare the query template with the data collected from the user
42             final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_3_b);
43             statement2.setString(1, "John");
44             statement2.setString(2, "Doe");
45             statement2.setInt(3, 20);
46
47             // Execute the populated query
48             final int rows_inserted = statement2.executeUpdate();
49             System.out.println(String.format("Done. %d row inserted in 'record process cost' table..", rows_inserted));
50
51             // Establish the connection with the database
52             connection = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=master;user=sa;password=123456");
53
54             // Prepare the query template with the data collected from the user
55             final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_c);
56             statement3.setString(1, "John");
57             statement3.setString(2, "Doe");
58             statement3.setInt(3, 20);
59
60             // Execute the populated query
61             final int rows_inserted = statement3.executeUpdate();
62             System.out.println(String.format("Done. %d row inserted in 'process' table..", rows_inserted));
63
64             // Get the user input of student ID
65             System.out.println("Please enter student ID:");
66             String studentId = sc.nextLine();
67
68             // Call nextInt() to consume the newline character, so that subsequent nextLine doesn't return nothing.
69             sc.nextInt();
70
71             // Get the user input of process ID
72             System.out.println("Please enter the process ID:");
73             String processId = sc.nextLine();
74
75             // Call nextInt() to consume the newline character, so that subsequent nextLine doesn't return nothing.
76             sc.nextInt();
77
78             // Get the user input of process Data
79             System.out.println("Please enter the process Data:");
80             String processData = sc.nextLine();
81
82             // Call nextInt() to consume the newline character, so that subsequent nextLine doesn't return nothing.
83             sc.nextInt();
84
85             // If process type is 1
86             if (processType == 1) {
87                 // Fit Process
88                 System.out.println("Please enter fit type:");
89                 String fitType = sc.nextLine();
90
91                 // Call nextInt() to consume the newline character, so that subsequent nextLine doesn't return nothing.
92                 sc.nextInt();
93
94                 // Get a database connection and prepare a query statement
95                 try (final Connection connection = DriverManager.getConnection(URL)) {
96                     final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_3_d);
97                     statement1.setString(1, "John");
98                     statement1.setString(2, "Doe");
99                     statement1.setInt(3, 20);
100                    statement1.setString(4, fitType);
101
102                    // Execute the populated query
103                    final int rows_inserted = statement1.executeUpdate();
104                    System.out.println(String.format("Done. %d row inserted in 'process' table..", rows_inserted));
105
106                    // Dispatching the query...*
107                    System.out.println("Dispatching the query...*");
108                    final int rows_inserted = statement1.executeUpdate();
109                    System.out.println(String.format("Done. %d row inserted..", rows_inserted));
110
111                } catch (SQLException e) {
112                    e.printStackTrace();
113                }
114            }
115
116            // If process type is 2
117            else if (processType == 2) {
118                // For Paint Process
119                System.out.println("Please enter Paint type:");
120                String paintType = sc.nextLine();
121
122                System.out.println("Please enter Paint method:");
123                String paintMethod = sc.nextLine();
124
125                // Call nextInt() to consume the newline character, so that subsequent nextLine doesn't return nothing.
126                sc.nextInt();
127
128                // Get a database connection and prepare a query statement
129                try (final Connection connection = DriverManager.getConnection(URL)) {
130
131                    final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_3_e);
132                    statement1.setString(1, "John");
133                    statement1.setString(2, "Doe");
134                    statement1.setInt(3, 20);
135                    statement1.setString(4, paintType);
136                    statement1.setString(5, paintMethod);
137
138                    // Execute the populated query
139                    final int rows_inserted = statement1.executeUpdate();
140                    System.out.println(String.format("Done. %d row inserted in 'process' table..", rows_inserted));
141
142                    // Dispatching the query...*
143                    System.out.println("Dispatching the query...*");
144                    final int rows_inserted = statement1.executeUpdate();
145                    System.out.println(String.format("Done. %d row inserted..", rows_inserted));
146
147                } catch (SQLException e) {
148                    e.printStackTrace();
149                }
150
151        }
152    }
153
154 }

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

443     }
444   }
445 }
446 }
447 }
448 }
449 } else {
450   // Information for Cat Process
451   System.out.println("Please enter the Cutting Type:");
452   final String cutType = sc.nextLine();
453   System.out.println("Please enter the Machine Type:");
454   final String machineType = sc.nextLine();
455   System.out.println("Connecting to the database... ");
456   System.out.println("Prepared a query statement");
457   try (final Connection connection = DriverManager.getConnection(URL)) {
458     try {
459       final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_3);
460       // Populate the query template with the data collected from the user
461       // /System.out.print("Name :");
462       statement1.setString1(sc.nextLine());
463       statement1.setFloat2(processed);
464       statement1.setString3(cutType);
465       statement1.setString4(machineType);
466       System.out.println("Dispatching the query...\"");
467       // Actually execute the populated query
468       final int rows_inserted = statement1.executeUpdate();
469       System.out.println(String.format("Done. %d rows inserted in \"%process\" table.", rows_inserted));
470     }
471     try {
472       final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
473       // Populate the query template with the data collected from the user
474       // /System.out.print("Name :");
475       statement3.setString1(processed);
476       statement3.setString2(cutType);
477       statement3.setString3(machineType);
478       System.out.println("Dispatching the query...\"");
479       // Actually execute the populated query
480       final int rows_inserted = statement3.executeUpdate();
481       System.out.println(String.format("Done. %d row inserted"));
482     }
483   }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
599 }
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

432     }
433   }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
599 }
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
589 }
```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

```

1  final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_B_C);
2  // Populate the query template with the data collected from the user
3  //System.out.println(name);
4  statement1.setInt(1, pName);
5  System.out.println("Displaying the query B(c)...");
6  //Actually execute the populated query
7  ResultSet rs = statement1.executeQuery();
8
9  while(rs.next()) {
10    System.out.println("Assembly ID: " + rs.getInt("assembly_id"));
11    System.out.println("Assembly account: " + rs.getString("assembly_account"));
12  }
13
14  final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_B_F);
15  // Populate the query template with the data collected from the user
16  //System.out.println(name);
17  statement2.setInt(1, assId);
18
19  System.out.println("Displaying the query B(f)...");
20  //Actually execute the populated query
21  ResultSet rs1 = statement2.executeQuery();
22  while(rs1.next()) {
23    float sup_cost = rs1.getFloat("sup_cost");
24    System.out.println("sup cost: " + sup_cost);
25  }
26
27  final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_B_I);
28
29  statement3.setInt(1, sup_cost);
30  System.out.println("Displaying query B(l)...");
31  //Actually execute the populated query
32  ResultSet rs2 = statement3.executeQuery();
33  while(rs2.next()) {
34    final int rows_inserted = statement3.executeUpdate();
35    System.out.println(String.format("Done. %d rows inserted in `Assembly_Account` table..", rows_inserted));
36  }
37
38  System.out.println("Process ID: " + pid);
39
40  try {
41    final PreparedStatement statement4 = connection.prepareStatement(QUERY_TEMPLATE_B_G);
42
43    //Populates the query template with the data collected from the user
44    //System.out.println(name);
45  }
46
47  catch (SQLException e) {
48    e.printStackTrace();
49  }
50
51  finally {
52    statement1.close();
53    statement2.close();
54    statement3.close();
55    statement4.close();
56  }
57
58  System.out.println("Department Account: " + procAcc);
59
60  try {
61    final PreparedStatement statement5 = connection.prepareStatement(QUERY_TEMPLATE_B_J);
62
63    statement5.setInt(1, procAcc);
64    System.out.println("Displaying query B(j)...");
65    //Actually execute the populated query
66    ResultSet rs3 = statement5.executeQuery();
67    while(rs3.next()) {
68      float sup_cost = rs3.getFloat("sup_cost");
69      System.out.println("sup cost: " + sup_cost);
70      sup_cost = sup_cost + tCost;
71    }
72
73    final PreparedStatement statement6 = connection.prepareStatement(QUERY_TEMPLATE_B_M);
74
75    statement6.setInt(1, sup_cost);
76    statement6.setInt(2, tCost);
77    System.out.println("Displaying query B(m)...");
78    final int rows_inserted = statement6.executeUpdate();
79    System.out.println(String.format("Done. %d rows inserted in `Process_Account` table..", rows_inserted));
80
81  }
82
83  finally {
84    statement5.close();
85  }
86
87  try {
88    final PreparedStatement statement7 = connection.prepareStatement(QUERY_TEMPLATE_B_D);
89
90    //Populates the query template with the data collected from the user
91    //System.out.println(name);
92    statements.setInt(1, pid);
93    System.out.println("Displaying the query B(d)...");
94    //Actually execute the populated query
95    ResultSet rs7 = statement7.executeQuery();
96    while(rs7.next()) {
97      float sup_cost = rs7.getFloat("sup_cost");
98      System.out.println("sup cost: " + sup_cost);
99      sup_cost = sup_cost + tCost;
100    }
101
102  }
103
104  finally {
105    statement7.close();
106  }
107
108  try {
109    final PreparedStatement statement8 = connection.prepareStatement(QUERY_TEMPLATE_B_E);
110
111    //Populates the query template with the data collected from the user
112    //System.out.println(name);
113    statements.setInt(1, pid);
114    System.out.println("Displaying query B(e)...");
115    //Actually execute the populated query
116    ResultSet rs8 = statement8.executeQuery();
117    while(rs8.next()) {
118      final int deptNo = rs8.getInt("dept_id");
119      System.out.println("Department ID: " + deptNo);
120    }
121
122  }
123
124  finally {
125    statement8.close();
126  }
127
128  try {
129    final PreparedStatement statement9 = connection.prepareStatement(QUERY_TEMPLATE_B_H);
130
131    statement9.setInt(1, deptNo);
132    System.out.println("Displaying query B(h)...");
133    //Actually execute the populated query
134    ResultSet rs9 = statement9.executeQuery();
135    while(rs9.next()) {
136      final String deptAcc = rs9.getString("dept_account");
137      System.out.println("Department Account: " + deptAcc);
138    }
139
140  }
141
142  finally {
143    statement9.close();
144  }
145
146  try {
147    final PreparedStatement statement10 = connection.prepareStatement(QUERY_TEMPLATE_B_K);
148
149    statement10.setInt(1, sup_cost);
150    statement10.setInt(2, deptNo);
151    System.out.println("Displaying query B(k)...");
152    //Actually execute the populated query
153    ResultSet rs10 = statement10.executeQuery();
154    while(rs10.next()) {
155      final float sup_cost = rs10.getFloat("sup_cost");
156      System.out.println("sup cost: " + sup_cost);
157    }
158
159  }
160
161  finally {
162    statement10.close();
163  }
164
165  try {
166    final PreparedStatement statement11 = connection.prepareStatement(QUERY_TEMPLATE_B_L);
167
168    statement11.setInt(1, jobNo);
169    System.out.println("Displaying query B(l)...");
170    //Actually execute the populated query
171    ResultSet rs11 = statement11.executeQuery();
172    while(rs11.next()) {
173      final int jobCompleted = rs11.getInt("job_completed");
174      System.out.println("Job completed: " + jobCompleted);
175    }
176
177  }
178
179  finally {
180    statement11.close();
181  }
182
183  System.out.println("Please enter the Kind of Job: 1. for Cut Job, 2 for Paint Job and 3 for Fit Job");
184
185  case "3":
186    break;
187  case "7":
188    System.out.println("Please enter the Job Number:");
189    final int jobNo = sc.nextInt();
190    sc.nextLine();
191
192    System.out.println("Please enter the date the job completed:");
193    final int jobCompleted = sc.nextInt();
194    sc.nextLine();
195
196    System.out.println("Please enter the Kind of Job: 1. for Cut Job, 2 for Paint Job and 3 for Fit Job");
197
198    break;
199  default:
200    System.out.println("Please enter the valid input");
201  }
202}

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help

```

```

1 package com.helloworld;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.util.Scanner;
9
10 public class HelloWorld {
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in);
13
14         System.out.println("Please enter the kind of Job: 1. for Cut Job, 2 for Paint Job and 3 for Fit Job:");
15         final int job_type = sc.nextInt();
16         sc.nextLine();
17
18         if (job_type == 1) {
19             System.out.println("Please enter the type of Machine used:");
20             final String machine_type = sc.nextLine();
21
22             System.out.println("Please enter the amount of time required in hours:");
23             final int timelabour = sc.nextInt();
24             sc.nextLine();
25
26             System.out.println("Please enter the labor time required in hours:");
27             final int laborestimate = sc.nextInt();
28             sc.nextLine();
29
30             System.out.println("Please enter the material used for the job:");
31             final String material = sc.nextLine();
32
33             System.out.println("Connecting to the database..");
34             // Get a database connection and prepare a query statement
35             try (final Connection connection = DriverManager.getConnection(DRIVER, URL)) {
36                 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_1_a);
37
38                 // Populate the query template with the data collected from the user
39                 //System.out.println("Name: " + name);
40                 statement1.setInt1(1, jobCompleted);
41                 statement1.setString1(2, machine_type);
42                 statement1.setInt1(3, laborestimate);
43                 statement1.setInt1(4, job_no);
44
45                 System.out.println("Dispatching the query ?(a)...");
46
47                 final int rows_inserted = statement1.executeUpdate();
48
49                 System.out.println("Done. " + rows_inserted + " rows inserted in 'CutJob' table..");
50             }
51
52         } else if (job_type == 2) {
53             System.out.println("Please enter the color of the paint used:");
54             final String color = sc.nextLine();
55
56             System.out.println("Please enter the value of paint used:");
57             final int paintvalue = sc.nextInt();
58             sc.nextLine();
59
60             System.out.println("Please enter the labor time in number of hours:");
61             final int laborestimate = sc.nextInt();
62             sc.nextLine();
63
64             System.out.println("Connecting to the database..");
65             // Get a database connection and prepare a query statement
66             try (final Connection connection = DriverManager.getConnection(DRIVER, URL)) {
67                 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_2_a);
68
69                 //System.out.println("Name: " + name);
70                 statement1.setInt1(1, jobCompleted);
71                 statement1.setString1(2, color);
72                 statement1.setInt1(3, laborestimate);
73                 statement1.setInt1(4, job_no);
74
75                 System.out.println("Dispatching the query ?(a)...");
76
77                 final int rows_inserted = statement1.executeUpdate();
78
79                 System.out.println("Done. " + rows_inserted + " rows inserted in 'Paint_Job' table..");
80             }
81
82         } else if (job_type == 3) {
83             System.out.println("Please enter the labor time in number of hours:");
84             final int laborestimate = sc.nextInt();
85             sc.nextLine();
86
87             System.out.println("Connecting to the database..");
88             // Get a database connection and prepare a query statement
89             try (final Connection connection = DriverManager.getConnection(DRIVER, URL)) {
90                 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_3_a);
91
92                 //System.out.println("Name: " + name);
93                 statement1.setInt1(1, jobCompleted);
94                 statement1.setInt1(2, laborestimate);
95
96                 System.out.println("Dispatching the query ?(a)...");
97
98                 final int rows_inserted = statement1.executeUpdate();
99
100                System.out.println("Done. " + rows_inserted + " rows inserted in 'Fit_Job' table..");
101            }
102        }
103    }
104 }

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

    //System.out.println("Department number");
    //sc.nextLine();
    System.out.println("Please enter department ID:");
    final int departmentnumber = sc.nextInt();
    // We call nextLine to consume that newline character, so that subsequent nextLine return nothing.
    //sc.nextLine();
    final String departmentdata = sc.nextLine(); // Read in user input of student First Name (white-spaces allowed).

    System.out.println("Connecting to the database..."); 
    try {
        final Connection connection = DriverManager.getConnection(url);
        final PreparedStatement statement = connection.prepareStatement(query_template_1);
        // Populate the query template with the data collected from the user
        statement.setInt(1, departmentnumber);
        statement.setString(2, departmentdata);
    } catch (SQLException e) {
        e.printStackTrace();
    }

    System.out.println("Dispatching the query..."); 
    // Actually execute the populated query
    final int row_inserted = statement.executeUpdate();
    System.out.println(String.format("%d row inserted.", row_inserted));
}

case '3':
    System.out.println("Please input the minimum category:");
    final int mincat = sc.nextInt();
    sc.nextLine();
    System.out.println("Please input the maximum category:");
    final int maxcat = sc.nextInt();
    sc.nextLine();
    System.out.println("Connecting to the database..."); 
    try {
        final Connection connection = DriverManager.getConnection(url);
        final PreparedStatement statement2 = connection.prepareStatement(query_template_2);
        statement2.setInt(1, mincat);
        statement2.setInt(2, maxcat);
        System.out.println("Dispatching the query 13..."); 
        ResultSet rs1 = statement.executeQuery();
        System.out.println("Content of the student table:");
        System.out.println("Name | Address | Category");
        // Unpack the tuples returned by the database and print them out to the user
        while (rs1.next()) {
            Writable
            SmartInsert
            512|31|28697
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

case '4':
    System.out.println("Please enter a new Assembly ID:");
    final int assembly_id = sc.nextInt();
    sc.nextLine();
    System.out.println("Please enter the Date ordered for the assembly:");
    final Date dateordered = sc.nextDate();
    sc.nextLine();
    System.out.println("Please enter assembly details:");
    final String assemblydetails = sc.nextLine();
    System.out.println("Please enter the associated Customer Name:");
    final String customername = sc.nextLine();
    System.out.println("Please enter the number of associated process IDs with this assembly");
    final int numassemblys = sc.nextInt();
    int[] assemblyarray = new int[numassemblys];
    for (int i = 0; i < numassemblys; i++) {
        System.out.println("Please enter " + i + "th " + " process ID:");
        final int processid = sc.nextInt();
        sc.nextLine();
    }

    try {
        final Connection connection = DriverManager.getConnection(url);
        final PreparedStatement statement4 = connection.prepareStatement(query_template_4_a);
        statement4.setInt(1, assembly_id);
        statement4.setString(2, assemblydetails);
        statement4.setString(3, customername);
        System.out.println("Dispatching the query 4(a)..."); 
        // actually execute the populated query
        final int row_inserted = statement4.executeUpdate();
        System.out.println(String.format("%d rows inserted in \'Assembly\' table.", row_inserted));
    } catch (SQLException e) {
        e.printStackTrace();
    }

    try {
        final PreparedStatement statement5 = connection.prepareStatement(query_template_4_b);
        // Populate the query template with the data collected from the user
        statement5.setInt(1, assembly_id);
        statement5.setInt(2, assembly_id);
        statement5.setString(1, assembly_customer);
        System.out.println("Dispatching the query 4(b)..."); 
        // Actually execute the populated query
        final int row_inserted = statement5.executeUpdate();
        System.out.println(String.format("%d rows inserted in \'order1\' table.", row_inserted));
    } catch (SQLException e) {
        e.printStackTrace();
    }

    try {
        final PreparedStatement statement6 = connection.prepareStatement(query_template_4_c);
        // Populate the query template with the data collected from the user
        statement6.setInt(1, assembly_id);
        for (int i = 0; i < numassemblys; i++) {
            statement6.setInt(2, assemblyarray[i]);
        }
        System.out.println("Dispatching the query 4(c)..."); 
        // Actually execute the populated query
        final int row_inserted = statement6.executeUpdate();
        System.out.println(String.format("%d rows inserted in \'passthrough\' table.", row_inserted));
    } catch (SQLException e) {
        e.printStackTrace();
    }

    case '5':
        System.out.println("Please enter the Job Number:");
        final int jobno = sc.nextInt();
        sc.nextLine();
        System.out.println("Please enter the date of job commenced:");
        final int datocommenced = sc.nextInt();
        sc.nextLine();
        System.out.println("Please enter the Process ID assigned to this job:");
        final int processid = sc.nextInt();
        sc.nextLine();
    }
}

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help

```

Code Snippet 1 (Line 1-185):

```

1 final int jobid = sc.nextInt();
2 sc.nextLine();
3 System.out.println("Please enter the Assembly ID assigned to the process for this job:");
4 final int assemblyId = sc.nextInt();
5 sc.nextLine();
6 try (final Connection connection = DriverManager.getConnection(URL)) {
7     final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_A_a);
8     statement1.setObject(1, jobid);
9     statement1.setObject(2, assemblyId);
10    statement1.executeUpdate();
11    System.out.println("Dispatching the query 6(a)...");
12    final int rows_inserted = statement1.executeUpdate();
13    System.out.println(String.format("Done. %d rows inserted in 'Job' table.", rows_inserted));
14 }
15
16 try (final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_A_b)) {
17     statement2.setObject(1, jobid);
18     statement2.setInt(2, assemblyId);
19     statement2.setInt(3, processId);
20     statement2.executeUpdate();
21     System.out.println("Dispatching the query 6(b)...");
22     final int rows_inserted = statement2.executeUpdate();
23     System.out.println(String.format("Done. %d rows inserted in 'assign' table.", rows_inserted));
24 }
25
26 case "4":
27     System.out.println("Please Enter the path of the file for importing data:");
28     sc.nextLine();
29     final String filePath = sc.nextLine();
30     final String line;
31     String line1 = "";
32     String splitBy = ",";
33     try {
34         br = new BufferedReader(new FileReader(filePath));
35     
```

Code Snippet 2 (Line 1-185):

```

1 String splitBy = ",";
2 try {
3     br = new BufferedReader(new FileReader(filePath));
4     try (final Connection connection = DriverManager.getConnection(URL)) {
5         while ((line = br.readLine()) != null) //returns a Boolean value
6         {
7             String[] customer = line.split(splitBy); // use comma as separator
8             //System.out.println("Customer Name - "+customer[0] + "\nAddress - " + customer[1] + "\ncategory - " + customer[2]);
9             try {
10                 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_B_a);
11                 statement1.setString(1, customer[0]);
12                 statement1.setString(2, customer[1]);
13                 statement1.setInt(3, Integer.parseInt(customer[2]));
14                 System.out.println("Dispatching the query 1..."); // execute the update command
15                 final int rows_inserted = statement1.executeUpdate();
16                 System.out.println(String.format("Done. %d rows inserted in 'customer' table.", rows_inserted));
17             }
18             }
19         br.close();
20     }
21     catch (FileNotFoundException e) {
22         e.printStackTrace(); //catch block
23         System.out.println("Could not find the file specified. Please try again!");
24     }
25     catch (IOException e) {
26         e.printStackTrace(); //IO Exception Ocuured! Please Try Again!
27     }
28 }
29
30 case "5":
31     System.out.println("Please enter the Department number for determining the labor cost:");
32     final int department = sc.nextInt();
33     sc.nextLine();
34     System.out.println("Please enter the date of completion for the jobs:");
35     final int dateCompleted = sc.nextInt();
36     sc.nextLine();
37     // Determining the Processes supervised by the given Department:
38     try (final Connection connection = DriverManager.getConnection(URL)) {
39         final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_ID_a));
40         // Populate the query Template with the data collected from the user
41     }
42 
```

Code Snippet 3 (Line 1-117):

```

117 final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_ID_a));
118 // Populate the query template with the data collected from the user
119 //System.out.println("Statement1 prepared successfully");
120 //statement1.executeUpdate();
121 System.out.println("Dispatching the query 1..."); // Actually executes the populated query
122 final Statement statement2 = connection.createStatement();
123 statement2.executeQuery();
124
125 int psize = 0;
126 int index = 0;
127 int[] prarray = new int[100];
128 int[] Arry = new int[100];
129 int jindex = 0;
130
131 while (rs.next()) {
132     int pid = rs.getInt("process_id");
133     prarray[index] = pid;
134     psize++;
135     index++;
136 }
137
138 for (int i = 0; i < psize; i++)
139 {
140     System.out.print(prarray[i]);
141 }
142
143 if (PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_ID_b))
144 {
145     statements.setInt1(prarray[1]);
146     System.out.println("Dispatching the query 3(b)..."); // execute the update command
147     final int numbers = 0;
148     while(rs.next())
149     {
150         int jindex = 0;
151         int id = rs.getInt("job_id");
152         Arry[jindex] = id;
153         jindex++;
154     }
155 }
156
157 int[] darray = new int[jindex];
158 int[] jarray = new int[jindex];
159 for (int k = 0; k < jindex; k++)
160 {
161     System.out.print(Arry[k]);
162 }
163
164 try {
165     final PreparedStatement statement4 = connection.prepareStatement(QUERY_TEMPLATE_ID_c));
166     statements.setInt1(jarray[1]);
167     System.out.println("Dispatching the query 3(c)..."); // execute the update command
168     final Statement rs1 = statement4.executeUpdate();
169     System.out.println(rs1);
170 }
171 
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

1168     while(rs2.next()) {
1169         tArray[1] = rs2.getInt("labor_time");
1170         dArray[1] = rs2.getInt("date_completed");
1171     }
1172 }
1173 int totalLaborTime = 0;
1174 for(int i=0; i<tArray.length; i++) {
1175     if(dArray[i] == dateCompleted) {
1176         totalLaborTime += tArray[i];
1177     }
1178 }
System.out.println("The total labor time within the given " + depth + " department for jobs completed during the " + dateCompleted + " date is: " + totalLaborTime + " hours.");
1179
1180     }
1181 }
1182
1183 case "11":
1184     System.out.print("Please enter the assembly ID through which the passed processes have to be determined:");
1185     final int abd = sc.nextInt();
1186     sc.nextLine();
1187     try (final Connection connection = DriverManager.getConnection(URL)) {
1188         try {
1189             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_11);
1190             statement1.setInt(1, abd);
1191             System.out.println("Dispatching the query 11...");
1192             ResultSet rs1 = statement1.executeQuery();
1193             System.out.println("Process ID Department Number:");
1194             while(rs1.next()) {
1195                 System.out.println(rs1.getInt(1) + "\t" + rs1.getInt(2));
1196             }
1197         }
1198     }
1199     break;
1200
1201 case "12":
1202     System.out.print("Please enter the date of completion of the job:");
1203     final int date = sc.nextInt();
1204     sc.nextLine();
1205     System.out.print("Please enter the Department for which the jobs are to be determined:");
1206     final int depo = sc.nextInt();
1207     sc.nextLine();
1208     try (final Connection connection = DriverManager.getConnection(URL)) {
1209         try {
1210             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_12_a);
1211             statement1.setTimestamp(1, new Timestamp(date));
1212             System.out.println("Dispatching the query 12(a)...");
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

1213             ResultSet rs1 = statement1.executeQuery();
1214             while(rs1.next()) {
1215                 System.out.println(rs1.getInt(1));
1216             }
1217         }
1218     }
1219     break;
1220
1221 case "13":
1222     System.out.print("Please enter the date of completion of the job:");
1223     final int date = sc.nextInt();
1224     sc.nextLine();
1225     System.out.print("Please enter the Department for which the jobs are to be determined:");
1226     final int depo = sc.nextInt();
1227     sc.nextLine();
1228     System.out.print("Connecting to the database...\n");
1229     try (final Connection connection = DriverManager.getConnection(URL)) {
1230         try {
1231             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_12_b);
1232             statement1.setInt(1, date);
1233             System.out.println("Dispatching the query 12(b)...");
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

1234             ResultSet rs1 = statement1.executeQuery();
1235             while(rs1.next()) {
1236                 System.out.println(rs1.getInt(1));
1237             }
1238         }
1239     }
1240     break;
1241
1242 case "14":
1243     System.out.print("Please enter the date of completion of the job:");
1244     final int date = sc.nextInt();
1245     sc.nextLine();
1246     System.out.print("Please enter the Department for which the jobs are to be determined:");
1247     final int depo = sc.nextInt();
1248     sc.nextLine();
1249     System.out.print("Connecting to the database...\n");
1250     try (final Connection connection = DriverManager.getConnection(URL)) {
1251         try {
1252             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_12_c);
1253             statement1.setInt(1, date);
1254             System.out.println("Dispatching query 12(c)...");
```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

1255             ResultSet rs1 = statement1.executeQuery();
1256             while(rs1.next()) {
1257                 System.out.println(rs1.getInt(1));
1258             }
1259         }
1260     }
1261     break;
1262
1263 case "15":
1264     System.out.print("Please enter the output file name (please don't give the extension):");
1265     final String fileName = sc.nextLine();
1266     System.out.print("Connecting to the database...\n");
1267     try (final Connection connection = DriverManager.getConnection(URL)) {
1268         try {
1269             final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_13);
1270             try {
1271                 statement2.setInt(1, cam);
1272                 System.out.println("Dispatching the query 13...");
1273                 ResultSet rs1 = statement2.executeQuery();
1274                 //System.out.println(rs1.getString("name") + "\t" + rs1.getString("address"));
1275                 try (PrintWriter writer = new PrintWriter(fileName + ".csv")) {
1276                     while(rs1.next()) {
1277                         String sb = new StringBuilder();
1278                         String name = rs1.getString("name");
1279                         sb.append(name);
1280                         sb.append(",");
1281                         String address = rs1.getString("address");
1282                         sb.append(address);
1283                         sb.append(",");
1284                         int category = rs1.getInt("category");
1285                         sb.append(category);
1286                         sb.append(",");
1287                         writer.println(sb.toString());
1288                         //System.out.println(name + "\t" + address + "\t" + cat);
1289                         //writer.println(name + "\t" + address + "\t" + cat);
1290                         //writer.println(name + "\t" + address + "\t" + cat);
1291                         //writer.println(name + "\t" + address + "\t" + cat);
1292                     }
1293                 } catch (FileNotFoundException e) {
1294                     System.out.println(e.getMessage());
1295                 }
1296             }
1297         }
1298     }
1299     break;
1300
1301 case "16":
1302     System.out.print("Please enter the minimum Job Number:");
1303     final int minJob = sc.nextInt();
1304
1305 }
```

The screenshot shows two code editors in the Eclipse IDE interface. Both windows have the title "eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE".

HelloWord.java:

```

1 package com;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class HelloWord {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Please enter the Job Number:");
10        int jobNo = sc.nextInt();
11        sc.nextLine();
12        System.out.println("Please enter the new color for the job:");
13        String jobColor = sc.nextLine();
14
15        Connection connection = DriverManager.getConnection("URL");
16        try {
17            final PreparedStatement statement1 = connection.prepareStatement("QUERY_TEMPLATE_14_a");
18            //Populate the query template with the data collected from the user
19            statement1.setInt(1, jobNo);
20            statement1.setString(2, jobColor);
21            System.out.println("Dispatching the query 14(a)...");
22            final int rows_inserted = statement1.executeUpdate();
23            System.out.println(String.format("Done, %d rows deleted in `cutjob` table..", rows_inserted));
24        } catch (Exception e) {
25            e.printStackTrace();
26        }
27
28        try {
29            final PreparedStatement statement2 = connection.prepareStatement("QUERY_TEMPLATE_14_b");
30            //Populate the query template with the data collected from the user
31            statement2.setInt(1, minNo);
32            statement2.setInt(2, maxNo);
33            System.out.println("Dispatching the query 14(b)...");
34            final int rows_inserted = statement2.executeUpdate();
35            System.out.println(String.format("Done, %d rows deleted in `assign` table..", rows_inserted));
36        } catch (Exception e) {
37            e.printStackTrace();
38        }
39
40        try {
41            final PreparedStatement statement3 = connection.prepareStatement("QUERY_TEMPLATE_14_c");
42            //Populate the query template with the data collected from the user
43            statement3.setInt(1, minNo);
44            statement3.setInt(2, maxNo);
45            System.out.println("Dispatching the query 14(c)...");
46            final int rows_inserted = statement3.executeUpdate();
47            System.out.println(String.format("Done, %d rows deleted in `job` table..", rows_inserted));
48        } catch (Exception e) {
49            e.printStackTrace();
50        }
51    }
52 }

```

AzureSQLDBC.java:

```

1 package com;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class AzureSQLDBC {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Please enter the minimum Job Number:");
10        int minNo = sc.nextInt();
11        sc.nextLine();
12        System.out.println("Please enter the maximum Job Number:");
13        int maxNo = sc.nextInt();
14        sc.nextLine();
15        System.out.println("Connecting to the database...\"");
16        Connection connection = DriverManager.getConnection("URL");
17        try {
18            final PreparedStatement statement1 = connection.prepareStatement("QUERY_TEMPLATE_14_a");
19            //Populate the query template with the data collected from the user
20            statement1.setInt(1, minNo);
21            statement1.setInt(2, maxNo);
22            System.out.println("Dispatching the query 14(a)...\"");
23            final int rows_inserted = statement1.executeUpdate();
24            System.out.println(String.format("Done, %d rows deleted in `cutjob` table..", rows_inserted));
25        } catch (Exception e) {
26            e.printStackTrace();
27        }
28
29        try {
30            final PreparedStatement statement2 = connection.prepareStatement("QUERY_TEMPLATE_14_b");
31            //Populate the query template with the data collected from the user
32            statement2.setInt(1, minNo);
33            statement2.setInt(2, maxNo);
34            System.out.println("Dispatching the query 14(b)...\"");
35            final int rows_inserted = statement2.executeUpdate();
36            System.out.println(String.format("Done, %d rows deleted in `assign` table..", rows_inserted));
37        } catch (Exception e) {
38            e.printStackTrace();
39        }
40
41        try {
42            final PreparedStatement statement3 = connection.prepareStatement("QUERY_TEMPLATE_14_c");
43            //Populate the query template with the data collected from the user
44            statement3.setInt(1, minNo);
45            statement3.setInt(2, maxNo);
46            System.out.println("Dispatching the query 14(c)...\"");
47            final int rows_inserted = statement3.executeUpdate();
48            System.out.println(String.format("Done, %d rows deleted in `job` table..", rows_inserted));
49        } catch (Exception e) {
50            e.printStackTrace();
51        }
52
53        case "15":
54            System.out.println("Please enter the Job Number for the color changer:");
55            int jobNo = sc.nextInt();
56            sc.nextLine();
57            System.out.println("Please enter the new color for the job:");
58            String jobColor = sc.nextLine();
59
60            Connection connection = DriverManager.getConnection("URL");
61            try {
62                final PreparedStatement statement1 = connection.prepareStatement("QUERY_TEMPLATE_15");
63                //Populate the query template with the data collected from the user
64                statement1.setInt(1, jobNo);
65                System.out.println("Dispatching the query 15...\"");
66                final int rows_inserted = statement1.executeUpdate();
67                System.out.println(String.format("Done, %d rows updated in `paintjob` table..", rows_inserted));
68            } catch (Exception e) {
69                e.printStackTrace();
70            }
71
72            break;
73
74        case "16":
75            System.out.println("Exiting! Good-bye!");
76            System.exit(0);
77            default:
78                System.out.println("Unrecognized option: " + option +
79                        "\nPlease try again.");
80                option();
81            break;
82
83        }
84    }
85 }

```

Snapshot showing successful compilation of the program (Terminal window shows program execution)

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

```

File Edit Source Refactor Search Project Run Window Help
Package Explorer Problems Javadoc Declaration Console
AzureSql src AzureSQLJDBC.java
JRE System Library [javaSE-12]
Referenced Libraries HelloWorld

1267     statement3.setInt(1, minJob);
1268     statement3.setInt(2, maxJob);
1269     System.out.println("Dispatching the query 14(c)...");
1270     // Actually execute the populated query
1271     final int rows_inserted = statement3.executeUpdate();
1272     System.out.println(String.format("Done. %d rows deleted in %s table...", rows_inserted));
1273     break;
1274 }
1275
1276 case "15":
1277     System.out.println("Please enter the Job Number for the color change:");
1278     final int jobno = sc.nextInt();
1279     sc.nextLine();
1280     System.out.println("Please enter the new color for the job:");
1281     final String color = sc.nextLine();
1282     System.out.println("Connecting to the database...");
1283     // Get the database connection, create statement and execute it right away, as no user input need be collected
1284     try (final Connection connection = DriverManager.getConnection(URL)) {
1285         try {
1286             final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_15);
1287             // Populate the query template with the data collected from the user
1288             //System.out.println(name);
1289             statement1.setString(1, color);
1290         }
1291     }
1292 }
1293
1294 case "16":
1295     System.out.println("Please enter customer name:");
1296     String name = sc.nextLine();
1297     System.out.println("Please enter customer address:");
1298     String address = sc.nextLine();
1299     System.out.println("Please enter customer category between 1 and 10:");
1300     int category = sc.nextInt();
1301     sc.nextLine();
1302     System.out.println("Connecting to the database...");
1303     JohnDispatcher.dispatchTheQuery...
1304     Done. 1 row inserted.
1305 }

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10)Retrieve total labor time within a Department for a given day;
11)Retrieve the process through which a given assembly-id has passed;
12)Retrieve the jobs completed during given date in a given department;
13)Retrieve the customers (in name order) whose category is in a given range;
14)Delete all cut-jobs with a Job Number;
15)Change the color of a Paint Job;
16)Import: enter new customers from a data file;
17)Export Customer Data
20) Exit!

```

AzureSQLJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 1:07:32 PM)
Welcome to the sample application!

Task 6. Java program Execution

Five queries for Type 1:

First:

eclipse-workspace - Eclipse IDE

```

File Edit Source Refactor Search Project Run Window Help
Package Explorer Problems Javadoc Declaration Console
AzureSql HelloWorld

1267     statement3.setInt(1, minJob);
1268     statement3.setInt(2, maxJob);
1269     System.out.println("Dispatching the query 14(c)...");

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10)Retrieve total labor time within a Department for a given day;
11)Retrieve the process through which a given assembly-id has passed;
12)Retrieve the jobs completed during given date in a given department;
13)Retrieve the customers (in name order) whose category is in a given range;
14)Delete all cut-jobs with a Job Number;
15)Change the color of a Paint Job;
16)Import: enter new customers from a data file;
17)Export Customer Data
20) Exit!

Please enter customer cname:
John
Please enter customer address:
London
Please enter customer category between 1 and 10:
1
Connecting to the database...
JohnDispatcher.dispatchTheQuery...
Done. 1 row inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10)Retrieve total labor time within a Department for a given day;
11)Retrieve the process through which a given assembly-id has passed;

```

AzureSQLJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 8:51:27 PM)
Welcome to the sample application!

Result:

The screenshot shows the Azure Data Studio interface. In the top bar, it says "SQLQuery_1 - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio". Below the bar, there's a toolbar with icons for File, Edit, View, Help, Welcome, Run, Disconnect, Change Connection, Explain, and Enable SQLCMD. The main area has a title bar "RESULTS" and "Messages". A table is displayed with columns "cname", "caddress", and "category". There is one row with values "John", "Norman", and "1". At the bottom, there are status bars for "Lr 1, Col 23", "Spaces: 4", "UTF-8", "LF", "SQL", "MSSQL", "1 rows", "00:00:00", and "gaur001-sql-server database.windows.net : cs-dsa-4513-sql-db".

Second:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". On the left, there's a "Package Explorer" view showing a project named "HelloWorld" with a file "HelloWorld.java". The code in "HelloWorld.java" contains a snippet of Java code using JDBC to interact with a database. The right side of the screen shows a terminal window with the output of the application. The terminal output includes a series of numbered options for interacting with a database, such as "Enter new customer", "Enter a new Transaction", etc., and some user input like "James" and "Norman". It also shows the execution of a query and the insertion of a new row into a table.

Result:

Azure Data Studio interface showing a query results table. The table has columns: cname, caddress, category. The data is:

cname	caddress	category
1 James	OKC	2
2 John	Norman	1

Third:

Eclipse IDE workspace showing a Java application named HelloWorld. The code in HelloWorld.java is:

```

statement3.setInt(1, minNo);
statements.setInt(2, maxNo);
System.out.println("Dispatching the query 14(c)...");

AzureSQLDB [Run Application] /usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full/linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 8:51:27 PM)
0) Enter a new Job;
1) Enter the Completion date of a Job;
2) Enter the Completion date of a Job;
3) Enter the Completion date of a Job;
4) Enter the Completion date of a Job;
5) Enter a new Department;
6) Enter a new Department;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve the total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export Customer Data;
20) Exit!
.
Please enter customer cname:
Steve
Please enter customer address:
.
Please enter customer category between 1 and 10:
Connecting to the database...
SteveDispatching the query...
Done. 1 row inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Transaction Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export Customer Data;
20) Exit!

```

Result:

Azure Data Studio interface showing a query results table. The table has columns: cname, address, and category. The data is as follows:

cname	address	category
1 James	OKC	2
2 John	Norman	1
3 Steve	Edmond	3

Fourth:

Eclipse IDE workspace showing a Java application named "HelloWorld". The code in HelloWorld.java includes a print statement: "System.out.println("Dispatching the query 14(c)...");". The Eclipse Console tab shows the output of the application's execution, which includes a series of numbered database queries and user input for some of them. The output ends with "Done. 2 rows inserted." and a list of options for the user to select.

```

AzureSQLDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jst.openjdt.hotspot/jre.full/linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 8:51:27 PM)
6) Enter a new Job:
7) Enter the Completion date of a job:
8) Enter a new Transaction Number:
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the process through which a given assembly-id has passed:
12) Retrieve the jobs completed during given dates in a given department:
13) Retrieve the customers (in name order) whose category is in a given range:
14) Delete all cut-jobs with a Job Number:
15) Change the color of a Paint Job:
16) Import: enter new customers from a data file:
17) Export: Customer Data
20) Exit!
Please enter customer name:
Smith
Please enter customer address:
Noble
Please enter customer category between 1 and 10:
2
Connecting to the database...
Done. 2 rows inserted.

Please select one of the options below:
1) Enter new customer
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly:
5) Enter a new Account Number:
6) Enter a new Transaction Number:
7) Enter the Completion date of a Job:
8) Enter a new Transaction Number:
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the process through which a given assembly-id has passed:
12) Retrieve the jobs completed during given dates in a given department:
13) Retrieve the customers (in name order) whose category is in a given range:
14) Delete all cut-jobs with a Job Number:
15) Change the color of a Paint Job:
16) Import: enter new customers from a data file:
17) Export: Customer Data
20) Exit!

```

Result:

The screenshot shows the Microsoft Azure Data Studio interface. In the top bar, it says "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". Below the bar, there's a "CONNECTIONS" section with a connection to "gaur0001-sql-server.database.windows.net". The main area has tabs for "Results" and "Messages". A SQL query "select * from Customer;" is run, and the results are displayed in a table:

	cname	caddress	category
1	James	OKC	2
2	John	Norman	1
3	Smith	Noble	4
4	Steve	Edmond	3

At the bottom, status information includes "Ln 1, Col 23 Spaces: 4 UTF-8 LF SQL MySQL 4 rows 00:00:00 gaur0001-sql-server.database.windows.net:cs-dsa-4513-sql-db".

Fifth:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". The left sidebar shows a "Package Explorer" with projects "AzureSql" and "HelloWorld". The main editor shows Java code for "HelloWorld.java":

```

statement3.setInt1( minNol);
statement3.setInt2( maxNol);
System.out.println("Dispatching the query 14(c)...");

1267
1268
1269

```

A "Terminal" view is open at the bottom, showing the output of a database query:

```

statement3.setInt1( minNol);
statement3.setInt2( maxNol);
System.out.println("Dispatching the query 14(c)...");

1267
1268
1269

Please enter customer cname:
Richard
Please enter customer address:
Moore
Please enter customer category between 1 and 10:
3
Connecting to the database...
RichardDispatching the query...
Done. 1 row inserted.

Please select one of the option below:
1) Enter a new customer;
2) Enter a new Department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction date;
9) Retrieve the total cost incurred on an assembly_id
10)Retrieve total labor time within a Department for a given day;
11)Retrieve the process through which a given assembly_id has passed;
12)Retrieve the jobs completed during given date in a given department;
13)Print the job completed during given date in a given department (order) whose category is in a given range;
14)Delete all cut-jobs with a Job Number;
15)Change the color of a Paint Job;
16)Import: enter new customers from a data file;
17)Export Customer Data
20) Exit!

```

Result

Azure Data Studio interface showing a query result. The query is:

```
1 select * from Customer
```

The results table has columns: cname, address, category. The data is:

cname	address	category
1 James	OKC	2
2 John	Norman	1
3 Richard	Moore	5
4 Smith	Noble	4
5 Steve	Edmond	3

Five queries of Type 2

First

Eclipse IDE workspace showing a Java application named "HelloWorld". The code in HelloWorld.java is:

```
statement1.setInt(1, minNo);
statement1.setInt(2, maxNo);
System.out.println("Dispatching the query 14(c)...");
```

The console output shows:

```
4) Enter a new Assembly:
5) Enter a new Account Number:
6) Enter a new Job:
7) Enter the Completion date of a Job:
8) Enter a new Transaction Number:
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the process through which a given assembly-id has passed:
12) Retrieve the jobs completed during given date in a given department:
13) Retrieve the customers (in name order) whose category is in a given range:
14) Delete all cut-jobs with a Job Number:
15) Change the color of a Paint Job:
16) Import: enter new customers from a data file:
17) Export: Customer Data
20) Exit!
2
Please enter department number
1
Please enter department data:
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new job;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the process through which a given assembly-id has passed:
12) Retrieve the jobs completed during given date in a given department:
13) Retrieve the customers (in name order) whose category is in a given range:
14) Delete all cut-jobs with a Job Number:
15) Change the color of a Paint Job:
16) Import: enter new customers from a data file:
17) Export: Customer Data
20) Exit!
```

Result

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

```
File Edit View Help
CONNECTIONS ...
Welcome SQLQuery_3.sql - gaur00...ur0001 SQLQuery_1 - gaur00...ur0001
Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD
1 select * from Department
```

Results

dept_id	dept_data
1	CS

Ln 1, Col 25 Spaces: 4 UTF-8 LF SQL MSSQL 1 rows 00:00:00 gaur0001-sql-server database.windows.net cs-dsa-4513-sql-db

Second:

Eclipse IDE workspace showing a Java project named "HelloWorld" and a terminal window output:

```
eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer HelloWorld.java
AzureSQL
HelloWorld
Problems Javadoc Declaration Compiler
AzureSQLDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 9:02:21 PM)
statement3.setInt(1, minNo);
statement3.setInt(2, maxNo);
System.out.println("Dispatching the query 14(c)...");

4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor cost for a department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cuts made with a job number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export Customer Data
20) Exit!
2
Please enter department number
10
Please enter department number:
10
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Job;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor cost for a department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cuts made with a job number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export Customer Data
20) Exit!
```

Result:

Azure Data Studio interface showing a query result. The query is:

```
select * from Departments
```

The results table is:

dept_id	dept_data
1	CS
2	ECE

Third:

Eclipse IDE workspace showing a Java project named "HelloWorld". The code in `HelloWorld.java` is:

```
statement3.setInt1(minNol);
statement3.setInt2(maxNol);
System.out.println("Dispatching the query 14(c)...");
```

The terminal window shows the following output:

```
4) Enter a new Assembly;
5) Enter a new Department Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Enter the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve all the jobs completed on a given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import or enter New customers from a data file;
17) Export Customer Data
20) Exit!
```

Then it prompts:

```
Please enter department number
33
Please enter department data:
javac
```

It then connects to the database and dispatches the query:

```
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted.
```

Finally, it lists the options again:

```
Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Department Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Enter the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve all the jobs completed on a given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import or enter New customers from a data file;
17) Export Customer Data
20) Exit!
```

Result:

Azure Data Studio interface showing a query results table. The table has columns 'dept_id' and 'dept_data'. The data is:

dept_id	dept_data
1	CS
2	ECE
3	PHY

Fourth:

Eclipse IDE workspace showing a Java project named 'HelloWorld'. The code in `HelloWorld.java` is:

```

1267     statement3.setInt(1, minNo);
1268     statement3.setInt(2, maxNo);
1269     System.out.println("Dispatching the query 14(c)...");

```

The terminal window shows the following interaction:

```

AzureSQLDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jdt.core/jre/bin/java [Nov 22, 2021, 9:02:21 PM]
4) Enter a new Assembly:
5) Enter a new Account Number:
6) Enter a new Department:
7) Enter the Completion date of a Job:
8) Enter a new Transaction Number:
9) Retrieve the cost incurred on an assembly-id
10)Retrieve total labor time within a Department for a given day:
11)Retrieve the process through which a given assembly-id has passed:
12)Retrieve the completion date of a given date in a given department:
13)Retrieves the customers (in name order) whose category is in a given range:
14)Delete all cut-jobs with a Job Number:
15)Change the color of a Paint Job:
16)Import: enter new customers from a data file:
17)Export Customer Data
20) Exit!
2>
Please enter department number
44
Please enter department data:
44
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted.

Please select one of the options below:
1)1)Enter a new Assembly:
2)Enter a new department;
3)Enter a new Process ;
4)Enter a new Assembly:
5)Enter a new Account Number:
6)Enter a new Job:
7)Enter the Completion date of a Job:
8)Enter a new Transaction Number:
9)Retrieve the total cost incurred on an assembly-id
10)Retrieve total labor time within a department for a given day:
11)Retrieve the process through which a given assembly-id has passed:
12)Retrieve the jobs completed during given date in a given department:
13)Retrieves the customers (in name order) whose category is in a given range:
14)Delete all cut-jobs with a Job Number:
15)Change the color of a Paint Job:
16)Import: enter new customers from a data file:
17)Export Customer Data
20) Exit!

```

Result:

Azure Data Studio interface showing a query results table. The table has two columns: dept_id and dept_data. The data is as follows:

dept_id	dept_data
1	CS
2	ECE
3	PHY
4	GEO

Fifth

Eclipse IDE workspace titled "eclipse-workspace". The "HelloWorld.java" file contains the following code:

```

1267     statements.setInt(1, minNo);
1268     statements.setInt(2, maxNo);
1269     System.out.println("Dispatching the query 14(c)...");

```

The terminal window shows the following log output:

```

ApacheJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jst.openjdt.hotspot/jre.full/linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 9:02:21 PM)
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export: Customer Data
20) Exit!
2
Please enter department number
25
Please enter department data:
HIEB
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the Completing date of a Job;
8) Enter a new Transaction Number;
9) Retrieve total cost incurred on an assembly-id
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Export: Customer Data
20) Exit!

```

Result:

The screenshot shows the Azure Data Studio interface. In the center, there's a results grid titled 'dept_data' with columns 'dept_id' and 'dept_name'. The data is as follows:

dept_id	dept_name
1	CS
2	ECE
3	PHV
4	GEO
5	CHEM

Ten queries for Type 3:

First

The screenshot shows the Eclipse IDE workspace. On the left, the package explorer shows a project named 'HelloWorld'. In the center, a Java file 'HelloWorld.java' is open, containing code related to JDBC. Below the editor, a terminal window displays the output of running the application. The terminal text includes:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
...
System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
...
trv (final Connection connection = DriverManager.getConnection(URL)) {
    ...
}
<terminated> AzureSQLDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jst/jdt/openjdk/hotspot/jre/full/linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 9:26:39 PM - 9:27:11 PM)
...
Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Employee;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Product;
7) Enter the Completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total cost incurred on an assembly-id
10) Retrieve all jobs taken within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the jobs completed during given date in a given department;
13) Insert a new job into a department whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import & export new customers from a data file;
17) Export Customer Data
20) Exit!
...
Please enter processId:
1
Please enter the process Data:
...
Please enter the Supervising Department ID:
11
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 For Cut Process:
...
Please enter Fit type:
1
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted in "process" table.
Please enter the query...
Done. 1 rows inserted.
Dispatching the query 3(d)...
Done. 1 rows inserted in "supervise" table.

```

Result:

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

```
File Edit View Help
CONNECTIONS
  servers
    > cs-dsa-4513-sql-db
      > Columns
        > job_id (PK, FK, int, not null)
        > labor_time (int, null)
        > Keys
        > Constraints
        > Triggers
        > Indexes
        > Statistics
      > dbo.RT_Process
        > Columns
          > process_id (PK, FK, int, not null)
          > process_name (varchar(255), null)
          > Keys
          > Constraints
          > Triggers
          > Indexes
          > Statistics
        > dbo.job
        > dbo.Movie
        > dbo.Orders
        > dbo.Paint_job
        > dbo.Paint_Process
        > dbo.PastThrough
        > dbo.Performer
        > dbo.Process
        > dbo.Process_Account
        > dbo.Record
        > dbo.Supervised_By
          > Columns
            > dept_id (PK, FK, int, not null)
            > dept_name (PK, FK, int, not null)
            > Keys
            > Constraints
      > dbo.Supervised_By
        > Columns
          > process_id (PK, FK, int, not null)
          > dept_id (PK, FK, int, not null)
        > Keys
        > Constraints
      > Triggers
      > Triggers
  AZURE
  > SQL SERVER BIG DATA CLUSTERS
  > < >
```

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute SQL/DQL

1 select * from Process

Results Messages

process_id	process_data
1	xyz

Line 1, Col 9 Spaces: 4 UTF-8 LF SQL MSQL 1 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db

• SQLQuery_3 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

```
File Edit View Help
CONNECTIONS
  servers
    > cs-dsa-4513-sql-db
      > Columns
        > job_id (PK, FK, int, not null)
        > labor_time (int, null)
        > Keys
        > Constraints
        > Triggers
        > Indexes
        > Statistics
      > dbo.RT_Process
        > Columns
          > process_id (PK, FK, int, not null)
          > process_name (varchar(255), null)
          > Keys
          > Constraints
          > Triggers
          > Indexes
          > Statistics
        > dbo.job
        > dbo.Movie
        > dbo.Orders
        > dbo.Paint_job
        > dbo.Paint_Process
        > dbo.PastThrough
        > dbo.Performer
        > dbo.Process
        > dbo.Process_Account
        > dbo.Record
        > dbo.Supervised_By
          > Columns
            > fit_type (PK, FK, int, not null)
            > Keys
            > Constraints
      > dbo.Supervised_By
        > Columns
          > process_id (PK, FK, int, not null)
          > dept_id (PK, FK, int, not null)
        > Keys
        > Constraints
      > Triggers
      > Triggers
  AZURE
  > SQL SERVER BIG DATA CLUSTERS
  > < >
```

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute SQL/DQL

1 select * from Fit_Type

Results Messages

process_id	fit_type
1	close

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 1 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

```
File Edit View Help
CONNECTIONS
  servers
    > cs-dsa-4513-sql-db
      > Columns
        > job_id (PK, FK, int, not null)
        > labor_time (int, null)
        > Keys
        > Constraints
        > Triggers
        > Indexes
        > Statistics
      > dbo.RT_Process
        > Columns
          > process_id (PK, FK, int, not null)
          > process_name (varchar(255), null)
          > Keys
          > Constraints
          > Triggers
          > Indexes
          > Statistics
        > dbo.job
        > dbo.Movie
        > dbo.Orders
        > dbo.Paint_job
        > dbo.Paint_Process
        > dbo.PastThrough
        > dbo.Performer
        > dbo.Process
        > dbo.Process_Account
        > dbo.Record
        > dbo.Supervised_By
          > Columns
            > dept_id (PK, FK, int, not null)
            > dept_name (PK, FK, int, not null)
            > process_id (PK, FK, int, not null)
            > Keys
            > Constraints
      > dbo.Supervised_By
        > Columns
          > process_id (PK, FK, int, not null)
          > dept_id (PK, FK, int, not null)
        > Keys
        > Constraints
      > Triggers
      > Triggers
  AZURE
  > SQL SERVER BIG DATA CLUSTERS
  > < >
```

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute SQL/DQL

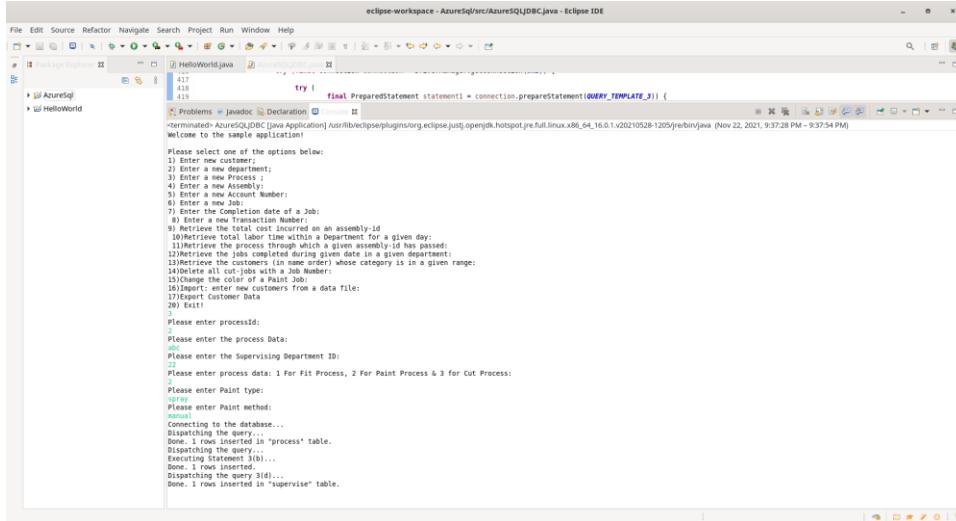
1 select * from Supervised_By

Results Messages

process_id	dept_id
1	11

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 1 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db

Second



```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer |  | HelloWord.java |  | src |  | 417 | 418 | 419 |
try {
    final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_3));
}

```

Problems = JavaDoc Declaration

-terminated-AzureSQLDBC [Java Application] Asuril/eclipseplugins/org.eclipse.jdt.core/jdt.hotspot/jre/full/linux.x86_64_16.0.1.v20210528-1205/jre/bin/java (Nov 22, 2021, 9:37:28 PM - 9:37:54 PM)

Welcome to the sample application!

Please select one of the options below:

- 1) Enter a new customer;
- 2) Enter a new department;
- 3) Enter a new Process ;
- 4) Enter a new Account ;
- 5) Enter a new Account Number;
- 6) Enter a new Job;
- 7) Enter a new Completion date of a Job;
- 8) Enter a new Transaction Number;
- 9) Retrieve all customers based on an assembly-Id;
- 10) Retrieve total labor time within a Department for a given day;
- 11) Retrieve the process through which a given assembly id has passed;
- 12) Retrieve the department through which a given department;
- 13) Retrieve the customers (in name order) whose category is in a given range;
- 14) Retrieve the account number;
- 15) Change the color of a Paint Job;
- 16) Import: enter new customers from a data file;
- 17) Import Customer Data;
- 20) Exit!

Please enter processId:

Please enter the process Data:

abc

Please enter the supervising Department ID:

2

Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 For Cut Process:

1

Please enter Paint type:

manual

Please enter Paint method:

manual

Connecting to the database...

Dispatching the query...

Done. 1 rows inserted in "process" table.

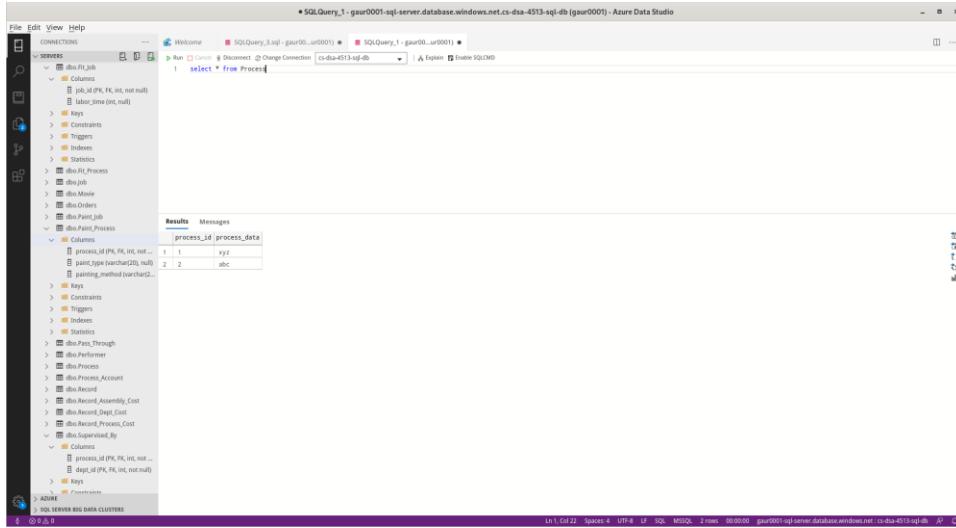
Dispatching the query...

Executing Statement 3(b):

Done. 1 rows inserted in "supervise" table.

Dispatching the query 3(d)... Done. 1 rows inserted in "supervise" table.

Result



File Edit View Help

CONNECTIONS

Welcome | SQLQuery_3.sql | gaur001..uri0001 | SQLQuery_1 - gaur001..uri0001 | Azure Data Studio

RESULTS

process_id process_data

process_id	process_data
1	xyz
2	abc

Messages

process_id process_data

1 xyz

2 abc

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

Welcome Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute TSQL

1 select * from Paint_Proc

Results Messages

process_id	paint_type	painting_method
1	spray	manual

process_id is PK, FK, int, not null
paint_type is nullable
painting_method is nullable

Keys Constraints Triggers Indexes Statistics

dbo.Paint_Proc

Columns

process_id, paint_type, painting_method

1 2

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

Welcome Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute TSQL

1 select * from Supervised_By

Results Messages

process_id	dept_id
1	11
2	22

process_id is PK, int, not null
dept_id is PK, FK, int, not null

Keys

process_id, dept_id

File Edit View Help

CONNECTIONS

• SQLQuery_3 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

Welcome Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db | Explain | Execute TSQL

1 select * from Paint_Proc

Results Messages

process_id	dept_id
1	11
2	22

process_id is PK, int, not null
dept_id is PK, FK, int, not null

Keys

process_id, dept_id

Third

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is as follows:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C) {
        // Populate the query template with the data collected from the user
        //System.out.println(cname);
        statement3.setString1(cname);
        statement3.setString2(cuttype);
        statement3.setString3(machineType);
        System.out.println("Dispatching the query...");
```

The output window shows the following sequence of events:

- 3) Enter a new Process :
- 4) Enter a new Customer :
- 5) Enter a new Account Number:
- 6) Enter a new Name:
- 7) Enter the Completion date of a Job:
- 8) Enter a new Transaction Number:
- 9) Enter a new Machine Type on an assembly-id
- 10)Retrieves total labor time within a Department for a given day:
- 11)Retrieves the tasks completed during a given date in a given department:
- 12)Retrieves the tasks completed during given date in a given department:
- 13)Retrieves the customers (in name order) whose category is in a given range:
- 14)Retrieves the customer's account number:
- 15)Change the color of a Paint Job:
- 16)Import: enter new customers from a data file:
- 17)Import: Customer Data
- 20) Exit!

Please enter processData:

```

Please enter the process Data:
def
```

Please enter the Supervising Department ID:
33

Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 For Cut Process:
1

Please enter the Cutting Type:
1

Please enter the Machine Type:
1

Connecting to the database...
Dispatching the query...
Done. 1 rows inserted
Dispatching the query...
Done. 1 rows inserted
Dispatching the query...
Done. 1 rows inserted
Done. 1 rows inserted in "supervise" table.

Result:

The screenshot shows the Microsoft SQL Server Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net-dba-4513-sql-db (gaur0001) - Azure Data Studio". The left sidebar shows the database structure for the "gaur0001-sql-server database window.net-dba-4513-sql-db" connection, including tables like "dbo.Account", "dbo.CutJob", and "dbo.Process".

The main pane displays the results of the query "select * from Process". The results table has two columns: "process_id" and "process_data". The data is as follows:

process_id	process_data
1	xyz
2	abc
3	def

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Cut_Process

Servers

- gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- cutting_type NVARCHAR(20),
- machine_type NVARCHAR(20),
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dev_Account
- dbo_Director
- dbo_Job
- Columns
- job_id PK, INT, NOT NULL
- labor_time INT, NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Sanders
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	cutting_type	machine_type
1	sharp	lathe

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 1 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

Servers

- gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- cutting_type NVARCHAR(20),
- machine_type NVARCHAR(20),
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dev_Account
- dbo_Director
- dbo_Job
- Columns
- job_id PK, INT, NOT NULL
- labor_time INT, NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Sanders
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	dept_id
1	11
2	22
3	33

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 3 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

Fourth:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSQL/src/AzureSQLDBC.java - Eclipse IDE". The code editor displays Java code for a "HelloWorld" application, specifically a section for preparing and executing SQL statements. Below the code editor is a terminal window showing the execution of the application. The terminal output includes:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    /*System.out.println("name");
    statement3.setString1(name);
    statement3.setString2(cuttype);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");*/
}

//terminated:AzureSQLDBC [Java Application] $java/eclipse/plugins/org.eclipse.jdt.core/openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/rebiny/java (Nov 22, 2021, 9:47:17 PM - 9:49:50 PM)
1) Enter a new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Product;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter a new location date of a Job;
8) Enter a new Transaction Number;
9) Enter a new Customer Data;
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve all customers whose birth date is a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a job number;
15) Insert a new Customer Data;
16) Import: enter new customers from a data file;
17) Update Customer Data;
20) Exit;

Please enter processId:
Please enter the process Data:
exit
Please enter the Supervising Department ID:
14
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 for Cut Process:
exit
Connecting to the database...
Dispatching the query...
Done. 1 rows inserted
Done. 1 rows inserted
Dispatching the query...
Done. 1 rows inserted
Done. 1 rows inserted in "supervise" table.

```

Result

The screenshot shows the Microsoft Data Studio interface with the title "SQLQuery_1 - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio". The left sidebar shows the database schema, including tables like "dbo.Account", "dbo.Cat", "dbo.Customer", and "dbo.Process". The main area shows the results of a SQL query:

```

SELECT * FROM Process

```

process_id	process_data
1	xyz
2	abc
3	def
4	seff

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...ur0001 SQLQuery_1 - gaur00...ur0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Fit_Process

RESULTS

process_id	fit_type
1	close
2	had

MESSAGES

1 L1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 2 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...ur0001 SQLQuery_1 - gaur00...ur0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

RESULTS

process_id	dept_id
1	11
2	22
3	33
4	44

MESSAGES

1 L1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 4 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Fifth:

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | HelloWord.java | 
try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    //System.out.println(cname);
    statement3.setString1(cname);
    statement3.setString2(custid);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");
}

```

Problems in Java or Declaration

-terminated:AzureSQLDBC [Java Application] /AzureSql/src/AzureSQLDBC.java full linux x86_64 16.0.1.v20210528-1205/rebin/java (Nov 22, 2021, 9:52:40 PM - 9:53:17 PM)

- 4) Enter a new Assembly:
- 5) Enter a new Job Number:
- 6) Enter a new Job:
- 7) Enter a new Completion date of a Job:
- 8) Enter a new Transaction Number:
- 9) Retrieves the total cost incurred in an assembly's ID
- 10) Retrieves the cost incurred in a department for a given day:
- 11)Retrieves the process through which a given assembly_id has passed:
- 12)Retrieves the customer whose last transaction date in a given department:
- 13)Retrieves the customers (in name order) whose category is in a given range:
- 14)Delete all cut_jobs with a Job Number:
- 15)Delete all paint_jobs with a Paint Job:
- 16)Import: enter new customers from a data file:
- 17)Export Customer Data
- 20) Exit

Please enter processId:
5
Please enter the process Data:
abc
Please enter the Supervising Department ID:
2
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 For Cut Process:
Please enter Paint type:
cost
Please enter Paint method:
manual
Connecting to the database...
Disconnecting the query.
Done. 1 rows inserted in "process" table.
Disconnecting the database...
Executing Statement 1(b)...
Done. 1 rows inserted
Disconnecting the database...
Done. 1 rows inserted in "supervise" table.

Result:

File Edit View Help

CONNECTIONS Welcome SQLQuery_1 - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio

gaur001-sql-server database window

Tables

- > dbo.Account
- > dbo.Arcel
- > dbo.Assembly
- > dbo.Assembly_Account
- > dbo.Assign
- > dbo.CutJob
- > dbo.Customer
- > dbo.FitJob
- > dbo.PaintJob
- > dbo.Process
- > > Columns
- > > process_id (PK, FK, int, not null)
- > > cutting_type (nchar(2), n, not null)
- > > machine_type (nchar(20), n, not null)
- > > Keys
- > > Constraints
- > > Triggers
- > > Indexes
- > > Statistics
- > > dbo.Department
- > > dbo.Employee
- > > dbo.Overtime
- > > dbo.Rate
- > > dbo.RjJob
- > > Columns
- > > job_id (PK, FK, int, not null)
- > > job_name (int, not null)
- > > Keys
- > > Constraints
- > > Triggers
- > > Indexes
- > > Statistics
- > > dbo.CutProcess
- > > dbo.FitProcess
- > > dbo.Job
- > > dbo.Movie
- > > dbo.Orders
- > > dbo.PaintJob
- > > dbo.PaintProcess

Results Messages

	process_id	process_data
1	1	xyz
2	2	abc
3	3	def
4	4	seff
5	5	sdf

Le 1 Col 22 Space: 4 UTF-8 LF SQL MODE: 5 rows 00:00:00 gaur001-sql-server database windows.net.cs-dsa-4513-sql-db

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001) - Azure Data Studio

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Paint_Process

Servers

- gaur0001-sql-server.database.windows.net
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id (PK, INT, NOT NULL)
- paint_type (VARCHAR(20), NOT NULL)
- machine_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Deg_Accnt
- dbo_Director
- dbo_Job
- Columns
- job_id (PK, INT, NOT NULL)
- labor_time (INT, NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Shoes
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	paint_type	painting_method
1	spray	manual
2	coat	manual

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 2 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001) - Azure Data Studio

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

Servers

- gaur0001-sql-server.database.windows.net
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id (PK, INT, NOT NULL)
- painting_type (VARCHAR(20), NOT NULL)
- machine_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Deg_Accnt
- dbo_Director
- dbo_Job
- Columns
- job_id (PK, INT, NOT NULL)
- labor_time (INT, NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Shoes
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	dept_id
1	11
2	22
3	33
4	44
5	55

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 5 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Sixth:

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    //System.out.println(cname);
    statement3.setString1(cname);
    statement3.setString2(cuttype);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");
```

-terminated-AzureSQLDBC [Java Application] Aswirile/eclipse/plugins/org.eclipse.jdt.core/openjdk.hotspot/jre/bin/java (Nov 22, 2021, 10:13:19 PM - 10:13:59 PM)

- 1) Enter a new customer;
- 2) Enter a new department;
- 3) Enter a new Process ;
- 4) Enter a new Order;
- 5) Enter a new Account Number;
- 6) Enter a new Job;
- 7) Enter a new Cut-Job;
- 8) Enter a new Transaction Number;
- 9) Enter a new Customer Data;
- 10) Retrieve total labor time within a Department for a given day;
- 11) Retrieve the process through which a given assembly-id has passed;
- 12) Retrieve all the customers whose birth date is a given department;
- 13) Retrieve the customers (in name order) whose category is in a given range;
- 14) Delete all cut-jobs with a job number;
- 15) Import: enter new Process Data;
- 16) Import: enter new customers from a data file;
- 17) Import: enter Customer Data;
- 20) Exit;

Please enter processId:

Please enter the process Data:

sql

Please enter the Supervising Department ID:

11

Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 for Cut Process:

process

Connecting to the database...

Dispatching the query...

Done. 1 rows inserted.

Done. 1 rows inserted.

Dispatching the query...

Done. 1 rows inserted.

Dispatching the query...

Done. 1 rows inserted in "supervise" table.

Result:

File Edit View Help

CONNECTIONS Welcome SQLQuery_3 - gaur001 - gaur001 - gaur001 SQLQuery_1 - gaur001 - gaur001 - gaur001 - Azure Data Studio

gaur001 - sql-server database window

Tables

- dbo.Account
- dbo.Arcel
- dbo.Assembly
- dbo.Assembly_Account
- dbo.Assign
- dbo.Cut_Job
- dbo.Customer
- dbo.Fit_Process
- dbo.Paint_Process
- dbo.Process
 - Columns
 - process_id (PK, FK, int, not null)
 - cutting_type (nchar(2), n, 0)
 - machine_type (nchar(20), n, 0)
 - Keys
 - process_id
 - constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Department
 - dbo.Employee
 - dbo.Director
 - dbo.Rt_Job
- dbo.Columns
 - job_id (PK, FK, int, not null)
 - user_name (int, not null)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics

- dbo.Fit_Process
- dbo.Job
- dbo.Movie
- dbo.Orders
- dbo.Paint_Job
- dbo.Paint_Process

Results Messages

process_id	process_data
1	xyz
2	abc
3	def
4	self
5	self
6	self

Le 1, Col 22 Space: 4 UTF-8 LF SQL MODE: 8-lines 00:00:00 gaur001 - sql-server database windows-apt -> dsa-4513.sql.db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..ur0001 SQLQuery_1 - gaur00..ur0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Fit_Process

RESULTS

process_id	fit_type
1	close
2	fed
3	smooth

MESSAGES

1

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..ur0001 SQLQuery_1 - gaur00..ur0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

RESULTS

process_id	dept_id
1	11
2	22
3	33
4	44
5	55
6	11

MESSAGES

1

Seventh:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    //System.out.println(cname);
    statement3.setString1(cname);
    statement3.setString2(cuttype);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");
}

```

The tooltip lists the following steps:

- 4) Enter a new Assembly:
- 5) Enter a new Cut Number:
- 6) Enter a new Job:
- 7) Enter a new Completion date of a Job:
- 8) Retrieves the total cost incurred on an assembly's ID
- 10) Retrieves the cost of a process which a Department for a given day:
- 11)Retrieves the process through which a given assembly's ID has passed:
- 12)Retrieves the customer's name whose birth date in a given department:
- 13)Retrieves the customer's (In name order) whose category is in a given range:
- 14)Delete all cut jobs with a Job Number:
- 15)Delete all paint jobs with a Job Number:
- 16)Import: enter new customers from a data file:
- 17)Export Customer Data
- 20) Exit

Please enter processId:
1
Please enter the process Data:
abc
Please enter the Supervising Department ID:
2
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 for Cut Process:
Please enter Paint type:
acrylic
Please enter Paint method:
brush
Connecting to the database...
Disconnecting the query...
Done. 1 rows inserted in "process" table.
Done. 1 rows inserted in "supervise" table.
Executing Statement 1(b)...
Done. 1 rows inserted
Executing Statement 2(a)...
Done. 1 rows inserted in "supervise" table.
Done. 1 rows inserted in "supervise" table.

Result:

process_id	process_data
1	xyz
2	abc
3	def
4	seff
5	sdf
6	eff
7	gh

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Paint_Process

Servers

- gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Audit
- dbo_Audit
- dbo_Audit
- dbo_Cut_Job
- dboCut_Process
- Columns
- process_id (PK, RI, INT, NOT NULL)
- paint_type (VARCHAR(20), NOT NULL)
- medium_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Movie
- dbo_Auditor
- dbo_Paint_Job
- dbo_Paint_Process

Results Messages

process_id	paint_type	painting_method
1	spray	manual
2	coat	manual
3	acrylic	brush

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 3 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

Servers

- gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Audit
- dbo_Audit
- dbo_Audit
- dbo_Cut_Job
- dboCut_Process
- Columns
- process_id (PK, RI, INT, NOT NULL)
- painting_type (VARCHAR(20), NOT NULL)
- medium_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Movie
- dbo_Auditor
- dbo_Paint_Job
- dbo_Paint_Process

Results Messages

process_id	dept_id
1	11
2	22
3	33
4	44
5	55
6	66
7	22

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 7 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Eight:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C) {
        // Populate the query template with the data collected from the user
        /*System.out.println(cname);
        statement3.setString1(cname);
        statement3.setString2(cuttype);
        statement3.setString3(machineType);
        System.out.println("Dispatching the query...");*/
    }
}

```

The output window displays a numbered list of 20 tasks, with the 13th task being executed:

- Enter a new Process :
- Enter a new Customer :
- Enter a new Account :
- Enter a new Transaction Number :
- Enter the completion date of a Job :
- Enter a new Job :
- Enter a new Assembly Number on an assembly-id :
- Retrieve total labor time within a Department for a given day :
- Retrieve the jobs completed during a given date in a given department :
- Retrieve the customers (in name order) whose category is in a given range :
- Change the color of a Paint Job :
- Import: enter new customers from a data file:
- Customer Data
- Exit!

Below this, the message "Please enter process Data:" is followed by a series of prompts and responses:

```

Please enter the process Data:
[1]
Please enter the Supervising Department ID:
[3]
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 For Cut Process:
[1]
Please enter the Cutting Type:
[fit]
Please enter the Cutting Type:
[fit]
Connecting to the database...
Dispatching the query...
Done, 1 rows inserted
Dispatching the query...
Done, 1 rows inserted
Dispatching the query...
Done, 1 rows inserted
[1]
[1]
[1]
[1]

```

Result

The screenshot shows the Microsoft SQL Server Data Studio interface with the title "SQLQuery_1 - gaur001-sql-server.database.windows.net-dsa-4513-sql-db (gaur001) - Azure Data Studio". The left sidebar shows the database structure for "gaur001-sql-server database windows.net-dsa-4513-sql-db". The main pane shows the results of the following query:

```

select * from Process

```

The results table contains 8 rows:

process_id	process_data
1	xyz
2	abc
3	def
4	seff
5	saf
6	eff
7	gfg
8	jkl

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001) - Azure Data Studio

Welcome Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Cut_Process

Servers

- gaur0001-sql-server.database.windows.net
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- cutting_type NVARCHAR(20), NOT NULL
- machine_type NVARCHAR(20), NOT NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Saw
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	cutting_type	machine_type
1	sharp	lathe
2	thick	knife

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 2 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

• SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001) - Azure Data Studio

Welcome Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

Servers

- gaur0001-sql-server.database.windows.net
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- cutting_type NVARCHAR(20), NOT NULL
- machine_type NVARCHAR(20), NOT NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Saw
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	dept_id
1	11
2	22
3	33
4	44
5	55
6	11
7	22
8	33

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 8 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Ninth

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is as follows:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    //System.out.println(cname);
    statement3.setString1(cname);
    statement3.setString2(cuttype);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");

    //terminated-AzureSQLDBC [Java Application] Aswirile/eclipse/plugins/org.eclipse.jdt.core/openJdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/rebin/java (Nov 22, 2021, 10:24:15 PM - 10:24:51 PM)
    1) Enter a new customer;
    2) Enter a new department;
    3) Enter a new Process ;
    4) Enter a new Product;
    5) Enter a new Account Number;
    6) Enter a new Job ;
    7) Enter a new location date of a Job;
    8) Enter a new Transaction Number;
    9) Enter a new Assembly Line;
    10) Retrieve total labor time within a Department for a given day;
    11) Retrieve the process through which a given assembly-id has passed;
    12) Retrieve all the products for a given date in a given department;
    13) Retrieve the customers (in name order) whose category is in a given range;
    14) Delete all cut-jobs with a job number;
    15) Import: enter new products from a file;
    16) Import: enter new customers from a data file;
    17) Import: enter Customer Data;
    20) Exit;

    Please enter processId:
    Please enter the process Data:
    xyz
    Please enter the Supervising Department ID:
    abc
    Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 for Cut Process:
    1
    Please enter Fit type:
    off
    Connecting to the database...
    Dispatching the query...
    Done. 1 rows inserted
    Dispatching the query...
    Done. 1 rows inserted
    Dispatching the query...
    Done. 1 rows inserted in "supervise" table.
  
```

Result:

The screenshot shows the Microsoft SQL Server Data Studio interface with the title "SQLQuery_1 - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio". The query run is:

```
select * from Process
```

The Results pane displays the following data:

process_id	process_data
1	xyz
2	abc
3	def
4	sdf
5	sdf
6	sdf
7	gfg
8	jkl
9	xyz

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Fit_Process

Servers

- gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_CutJob
- dbo_Cut_Process
- Columns
- process_id (PK, INT, NOT NULL)
- fit_type (nchar(20), NOT NULL)
- machine_type (nchar(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dir_Account
- dbo_Director
- dbo_Job
- dbo_Customer
- dbo_CutJob
- dbo_Cut_Process
- Columns
- yo_id (PK, INT, NOT NULL)
- labor_time (INT, NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Sabotage
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	fit_type
1	close
2	bed
3	smooth
4	soft

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 4 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_R

Servers

- gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_CutJob
- dbo_Cut_Process
- Columns
- process_id (PK, INT, NOT NULL)
- fit_type (nchar(20), NOT NULL)
- machine_type (nchar(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dir_Account
- dbo_Director
- dbo_Job
- dbo_Customer
- dbo_CutJob
- dbo_Cut_Process
- Columns
- yo_id (PK, INT, NOT NULL)
- labor_time (INT, NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_job
- dbo_Mouse
- dbo_Sabotage
- dbo_Paint_Job
- dbo_Paint_Process

AZURE

SQL SERVER BIG DATA CLUSTERS

Results Messages

process_id	dept_id
1	11
2	22
3	33
4	44
5	55
6	66
7	77
8	88
9	99

Ln 1, Col 28 Spaces: 4 UTF-8 LF SQL MSQL 9 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

Tenth:

```

try {
    final PreparedStatement statement3 = connection.prepareStatement(QUERY_TEMPLATE_3_C);
    // Populate the query template with the data collected from the user
    //System.out.println(cname);
    statement3.setString1(cname);
    statement3.setString2(processId);
    statement3.setString3(machineType);
    System.out.println("Dispatching the query...");
}

```

The tooltip lists the following methods:

- 4) Enter a new Assembly:
- 5) Enter a new Customer Number:
- 6) Enter a new Job:
- 7) Enter a new Completion date of a Job:
- 8) Enter a new Transaction Number:
- 9) Retrieves the total cost incurred on an assembly's ID
- 10) Retrieves the cost of a process which a department has passed:
- 11)Retrieves the process through which a given assembly-id has passed:
- 12)Retrieves the customer name whose completion date in a given department:
- 13)Retrieves the customer s (In name order) whose category is in a given range:
- 14)Delete all cut jobs with a Job Number:
- 15)Delete all paint jobs with a Job Number:
- 16)Import: enter new customers from a data file:
- 17)Export Customer Data
- 20) Exit

Please enter processId:
10
Please enter the process Data:
xyz
Please enter the Supervising Department ID:
20
Please enter process data: 1 For Fit Process, 2 For Paint Process & 3 for Cut Process:
Please enter Paint type:
Spray
Please enter Paint method:
Manual
Connecting to the database...
Disconnecting the query.
Done, 1 rows inserted in "process" table.
Disconnecting the database...
Executing Statement 1(b)...
Done, 1 rows inserted
Disconnecting the database...
Done, 1 rows inserted in "supervise" table.

Results:

process_id	process_data
1	xyz
2	abc
3	def
4	sef
5	ufd
6	ef
7	gf
8	jk
9	yui
10	tre

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Paint_Process

Servers

- gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- paint_type VARCHAR(20), NOT NULL
- machine_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dir_Account
- dbo_Director
- dbo_Director_Job
- dbo_Colour
- job_id PK, FK, INT, NOT NULL
- labor_time INT, NOT NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_Job
- dbo_Movie
- dbo_Schedule
- dbo_Paint_Job
- dbo_Paint_Process

Results Messages

process_id	paint_type	painting_method
1	spray	manual
2	coat	manual
3	acrylic	brush
4	spray	manual

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSSQL 4 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Supervised_H

Servers

- gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db (gaur0001)
- dbo
- dbo_Audit
- dbo_Assemblies
- dbo_Assembly_Account
- dbo_Audit
- dbo_AuditConsumer
- dbo_Cut_Job
- dbo_Cut_Process
- Columns
- process_id PK, INT, NOT NULL
- painting_type VARCHAR(20), NOT NULL
- machine_type (VARCHAR(20), NOT NULL)
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Department
- dbo_Dir_Account
- dbo_Director
- dbo_Director_Job
- dbo_Colour
- job_id PK, FK, INT, NOT NULL
- labor_time INT, NOT NULL
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- dbo_Pt_Process
- dbo_Job
- dbo_Movie
- dbo_Schedule
- dbo_Paint_Job
- dbo_Paint_Process

Results Messages

process_id	dept_id
1	11
2	22
3	33
4	44
5	55
6	66
7	77
8	88
9	99
10	55

Line 1, Col 28 Spaces: 4 UTF-8 LF SQL MSSQL 10 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Ten Queries of Type 4:

First:

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Search Project Run Window Help
# | Package Explorer | 0 HelloWorld.java | 1 |
71     case "4":
72         System.out.println("Please enter a new Assembly ID:");
73
74         String assemblyId = scanner.nextLine();
75
76         System.out.println("Please select one of the options below:");
77         System.out.println("1) Enter a new department");
78         System.out.println("2) Enter a new process");
79         System.out.println("3) Enter a new Account Number");
80         System.out.println("4) Enter a new Customer Name");
81         System.out.println("5) Enter the Completion date of a Job");
82         System.out.println("6) Enter a new Transaction cost incurred on an assembly Id");
83         System.out.println("7) Retrieve total labor time within a Department for a given day");
84         System.out.println("8) Retrieve the number of hours worked by an employee if has passed");
85         System.out.println("9) Retrieve the jobs completed during given date in a given department");
86         System.out.println("10) Retrieve the customers (in name order) whose category is in a given range");
87         System.out.println("11) Retrieve cut-job");
88         System.out.println("12) Change the color of a Paint Job");
89         System.out.println("13) Insert: enter new customers from a data file");
90         System.out.println("14) Export: customer data");
91         System.out.println("15) Exit");
92
93         scanner.nextLine();
94
95         System.out.println("Please enter a new Assembly ID:");
96
97         assemblyId = scanner.nextLine();
98
99         System.out.println("Please enter the Date ordered for the assembly:");
100        Date orderedDate = scanner.nextDate();
101
102        System.out.println("Please enter the Assembly details:");
103        String assemblyDetails = scanner.nextLine();
104
105        System.out.println("Please enter the associated Customer Name:");
106        String customerName = scanner.nextLine();
107
108        System.out.println("Please enter the number of associated process IDs with this assembly");
109        int numProcesses = scanner.nextInt();
110
111        System.out.println("Please enter 0th process ID:");
112        String processId0 = scanner.nextLine();
113
114        System.out.println("Please enter 1th process ID:");
115        String processId1 = scanner.nextLine();
116
117        System.out.println("Dispatching the query 4(a)...");
118        String query4a = "INSERT INTO Assembly (assembly_id, assembly_name, assembly_type, assembly_color, assembly_weight, assembly_size, assembly_cost, assembly_completion_date, assembly_details, assembly_customer_name, assembly_processes) VALUES (" + assemblyId + ", '" + assemblyName + "', '" + assemblyType + "', '" + assemblyColor + "', " + assemblyWeight + ", " + assemblySize + ", " + assemblyCost + ", '" + orderedDate + "', '" + assemblyDetails + "', '" + customerName + "', " + numProcesses + ");";
119
120        Statement statement4a = connection.createStatement();
121        statement4a.executeUpdate(query4a);
122
123        System.out.println("Done. " + numProcesses + " rows inserted in " + "Assembly" + " table.");
124
125        System.out.println("Dispatching the query 4(b)...");
126        String query4b = "INSERT INTO OrderLine (order_id, quantity, unit_price, line_total, line_order) VALUES (" + processId0 + ", " + quantity + ", " + unitPrice + ", " + lineTotal + ", " + lineOrder + ");";
127
128        Statement statement4b = connection.createStatement();
129        statement4b.executeUpdate(query4b);
130
131        System.out.println("Done. 1 rows inserted in " + "orderline" + " table.");
132
133        System.out.println("Dispatching the query 4(b)...");
134        String query4c = "INSERT INTO PaintThrough (process_id, paint_id, quantity, unit_price, line_total, line_order) VALUES (" + processId1 + ", " + paintId + ", " + quantity + ", " + unitPrice + ", " + lineTotal + ", " + lineOrder + ");";
135
136        Statement statement4c = connection.createStatement();
137        statement4c.executeUpdate(query4c);
138
139        System.out.println("Done. 1 rows inserted in " + "passthrough" + " table.");
140
141        scanner.nextLine();
142
143        System.out.println("Please select one of the options below:");
144
145    }

```

Result:

• SQLQuery_1 - gaur0001-sql-server.database.windows.net-dba-4513-sql-db (gaur0001) - Azure Data Studio

assembly_id	assembly_details	date_ordered
fgh		10/10/2011

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Order

Results Messages

name	assembly_id
John	1

Ln 1, Col 21 Spaces: 4 LF SQL MSOLK 1 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur001..uri0001 SQLQuery_1 - gaur00..uri0001

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

Results Messages

assembly_id	process_id
1	1
2	2

Ln 1, Col 8 Spaces: 4 LF SQL MSOLK 2 rows 00:00:08 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db

Second:

```

eclipse-workspace - AzureSql\src\AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | src | AzureSql | HelloWorld |
src | AzureSQLJDBC.java | 671 | 672 | System.out.println("Please enter a new Assembly ID:");
673 | Problems = JavaDoc Declaration Comments |
674 | AzuresSQLJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.linux.x86_64_16.0.1/v20210526-1205/rebir/java (Nov 22, 2021, 10:47:33 PM)
675 | 4) Enter a new Assembly:
676 | 5) Enter a new Account Number:
677 | 6) Enter a new Customer Name:
678 | 7) Enter the Completion date of a Job:
679 | 8) Enter the Date ordered:
680 | 9) Retrieve the total cost incurred on an assembly_id
681 | 10)Retrieve total labor cost within a Department for a given day:
682 | 11)Retrieve the department id for which a job has passed:
683 | 12)Retrieve the jobs completed during given date in a given department:
684 | 13)Retrieve the job (job_id, order_id) whose category is in a given range:
685 | 14)Delete all cut-jobs with a Job Number:
686 | 15)Change the color of a Paint Job:
687 | 16)Delete all customers from a data file:
688 | 17)Export Customer Data:
689 | 20) Exit!
690 |
691 | Please enter a new Assembly ID:
692 | 1
693 | Please enter the Date ordered for the assembly:
694 | 20210526
695 | Please enter the Assembly details:
696 | 1
697 | Please enter the associated Customer Name:
698 | Smith
699 | Please enter the number of associated process IDs with this assembly
700 | 3
701 | Please enter 1th process ID:
702 | 1
703 | Please enter 2th process ID:
704 | 2
705 | Dispatching the query 4(a)...
706 | Done. 1 rows inserted in "Assembly" table.
707 | Dispatching the query 4(b)...
708 | Done. 1 rows inserted in "order1" table.
709 | Dispatching the query 4(c)...
710 | Done. 1 rows inserted in "assembly" table.
711 | Dispatching the query 4(d)...
712 | Done. 1 rows inserted in "passthrough" table.
713 | Dispatching the query 4(e)...
714 | Done. 1 rows inserted in "passthrough" table.
715 | Dispatching the query 4(f)...
716 | Done. 1 rows inserted in "passthrough" table.
717 |
718 | All done. See the bottom for the output below.

```

Results:

assembly_id	assembly_details	date_ordered
1	rgh	10102011
2	tyu	10152020

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...ur0001 SQLQuery_1 - gaur00...ur0001

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Order

Results Messages

name	assembly_id
John	1
Smith	2

job_id (PK, FK, int, not null)
labor_time (int, null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.RT_Process
dbo.ODL
dbo.Movie
dbo.Orders
Columns
c_name (PK, FK, varchar(20),
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
paint_type (varchar(20), null)
posting_method (varchar(20))
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Paint_Process
Columns
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Orders
Columns
c_name (PK, FK, varchar(20),
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
paint_type (varchar(20), null)
posting_method (varchar(20))
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Paint_Process
Columns
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
Keys
Constraints
Triggers

AZURE

SQL SERVER BIG DATA CLUSTERS

Ln 1, Col 21 Spaces: 4 LF SQL MSOLK 2 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...ur0001 SQLQuery_1 - gaur00...ur0001

Run Cancel Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

Results Messages

assembly_id	process_id
1	1
2	1
2	2
4	2
5	2

job_id (PK, FK, int, not null)
labor_time (int, null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.RT_Process
dbo.ODL
dbo.Movie
dbo.Orders
Columns
c_name (PK, FK, varchar(20),
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
paint_type (varchar(20), null)
posting_method (varchar(20))
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Orders
Columns
c_name (PK, FK, varchar(20),
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
paint_type (varchar(20), null)
posting_method (varchar(20))
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Orders
Columns
c_name (PK, FK, varchar(20),
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
paint_type (varchar(20), null)
posting_method (varchar(20))
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Orders
Columns
assembly_id (PK, FK, int, not null)
process_id (PK, FK, int, not null)
Keys
Constraints
Triggers

AZURE

SQL SERVER BIG DATA CLUSTERS

Ln 1, Col 27 Spaces: 4 LF SQL MSOLK 5 rows 00:00:00 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Third:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is:

```

public class HelloWorld {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Please enter a new Assembly ID:");
        // 1
        // 2
        // 3
        // 4
        // 5
        // 6
        // 7
        // 8
        // 9
        // 10
        // 11
        // 12
        // 13
        // 14
        // 15
        // 16
        // 17
        // 18
        // 19
        // 20
    }
}

```

The terminal window shows the following output:

```

Please enter a new Assembly ID:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20) Exit!
Please enter a new Assembly ID:
1
Please enter the Date ordered for the assembly:
10/12/2020
Please enter the Assembly details:
Assembly ID:
Please enter the associated Customer Name:
Janes
Please enter the number of associated process IDs with this assembly
3
Please enter 1th process ID:
Please enter 1th process ID:
Please enter 2th process ID:
Please enter 2th process ID:
Dispatching the query 4(a)...
Done. 1 rows inserted in "Assembly" table.
Dispatching the query 4(b)...
Done. 1 rows inserted in "order1" table.
Dispatching the query 4(b)...
Done. 1 rows inserted in "passthrough" table.
Dispatching the query 4(b)...
Done. 1 rows inserted in "passthrough" table.
Dispatching the query 4(b)...
Done. 1 rows inserted in "passthrough" table.
Done. 1 rows inserted in "passthrough" table.
Please enter one of the options below:

```

Results:

The screenshot shows the Azure Data Studio interface with the title "SQL Query_1 - gaur0001-sql-server.database.windows.net-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The query run is:

```

select * from Assembly

```

The results pane shows the following data:

assembly_id	assembly_name	date_ordered
1	rgh	10/12/2020
2	tyu	10/12/2020
3	qwe	10/12/2020

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...urn0001 SQLQuery_1 - gaur00...urn0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Order

Results Messages

name	assembly_id
James	3
John	1
Smith	2

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...urn0001 SQLQuery_1 - gaur00...urn0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

Results Messages

assembly_id	process_id
1	1
2	2
3	3
4	4
5	5
6	3
7	3
8	7

File Edit View Help

CONNECTIONS

Welcome SQLQuery_3.sql - gaur00...urn0001 SQLQuery_1 - gaur00...urn0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

Results Messages

assembly_id	process_id
1	1
2	2
3	3
4	4
5	5
6	3
7	3
8	7

Fourth:

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | Java Resources | src | AzureSql | HelloWorld | HelloWord.java | 100%
871    case "4":
872        System.out.println("Please enter a new Assembly ID:");
873
874        Problems = JavaDocDeclaration();
875
876        AzureSQLDBC.java Application /src/librecycle/plugins/org.eclipse.justjavajdk.hotspot.jre.full/linux.x86_64_16.0.1/v20210526-1205/rebir/java (Nov 22, 2021, 10:47:33 PM)
877
878        Please select one of the options below:
879        1) Enter new customer;
880        2) Enter a new Account;
881        3) Enter a new Process ;
882        4) Enter a new Assembly;
883        5) Enter a new Account Number;
884        6) Enter a new Job;
885        7) Enter a completion date of a Job;
886        8) Enter a new Transaction Number;
887        9) Enter a new Order number on an assembly_id;
888        10)Retrieve total labor time within a given day;
889        11)Retrieve the process through which a given assembly id has passed;
890        12)Enter a new customer number;
891        13)Enter the customers (in name order) whose category is in a given range;
892        14)Enter the cut-job of a Job Number;
893        15)Change the cut-job of a Job Number;
894        16)Import: enter new customers from a data file;
895        17)Import: Customer Data;
896        20) Exit!
897
898        Please enter a new Assembly ID;
899        Please enter the Date ordered for the assembly:
900        11102013
901        Please enter the Assembly details:
902        unt
903        Please enter the associated Customer Name:
904        unt
905        Please enter the number of associated process IDs with this assembly
906        Please enter 6th process ID;
907        Please enter 1th process ID;
908        10
909        Dispatching the query 4...
910        Done, 1 rows inserted in "Assembly" table.
911        Dispatching the query 4(b)...
912        Done, 1 rows inserted in "Customer" table.
913        Dispatching the query 4(b)...
914        Done, 1 row inserted in "passthrough" table.
915        Dispatching the query 4...
916        Done, 1 rows inserted in "passthrough" table.
917
918        Please select one of the options below:

```

Results:

File Edit View Help

CONNECTIONS

Welcome SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

SQL Query_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)

Run Document Change Connection ts-dsa-4513-sql-db Export Enable SQLCMD

1. select * from Assembly;

assembly_id	assembly_details	date_ordered
1	rgh	10102011
2	tyu	10152020
3	qwe	10102020
4	wer	11102013

File Edit View Help

servers

File Edit View Help

CONNECTIONS

Welcome SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

SQL Query_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)

Run Document Change Connection ts-dsa-4513-sql-db Export Enable SQLCMD

1. select * from Assembly;

assembly_id	assembly_details	date_ordered
1	rgh	10102011
2	tyu	10152020
3	qwe	10102020
4	wer	11102013

File Edit View Help

servers

File Edit View Help

CONNECTIONS

Welcome SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

SQL Query_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)

Run Document Change Connection ts-dsa-4513-sql-db Export Enable SQLCMD

1. select * from Assembly;

assembly_id	assembly_details	date_ordered
1	rgh	10102011
2	tyu	10152020
3	qwe	10102020
4	wer	11102013

File Edit View Help

servers

File Edit View Help

CONNECTIONS

Welcome SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

SQL Query_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)

Run Document Change Connection ts-dsa-4513-sql-db Export Enable SQLCMD

1. select * from Assembly;

assembly_id	assembly_details	date_ordered
1	rgh	10102011
2	tyu	10152020
3	qwe	10102020
4	wer	11102013

File Edit View Help

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Execute TSQL

1 select * from Order

CONNECTIONS

services

- job_id (PK, FK, int, not null)
 - > Keys
 - > Constraints
 - > Triggers
 - > Indexes
 - > Statistics
 - > db.DP_Procedures
 - > db.DL_Procedures
 - > db.DM_Procedures
 - > dbo.Orders
 - > Columns
 - name (PK, FK, varchar(20), not null)
 - assembly_id (PK, FK, int, not null)
 - process_id (PK, FK, int, not null)
 - paint_type (varchar(20), null)
 - painting_method (varchar(20), null)
 - > Keys
 - > Constraints
 - > Triggers
 - > Indexes
 - > Statistics
 - > db.DP_Procedures
 - > db.DL_Procedures
 - > db.DM_Procedures
 - > db.DP_Procedures
 - > db.DL_Procedures
 - > db.DM_Procedures

Results Messages

name	assembly_id
James	3
John	1
Richard	4
Smith	2

Line 1, Col 21 Spaces: 4 UTF-8 LF SQL MSOL 4 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db

File Edit View Help

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

CONNECTIONS

services

- job_id (PK, FK, int, not null)
 - > Keys
 - > Constraints
 - > Triggers
 - > Indexes
 - > Statistics
 - > db.DP_Procedures
 - > db.DL_Procedures
 - > db.DM_Procedures
 - > dbo.Orders
 - > Columns
 - name (PK, FK, varchar(20), not null)
 - assembly_id (PK, FK, int, not null)
 - process_id (PK, FK, int, not null)
 - paint_type (varchar(20), null)
 - painting_method (varchar(20), null)
 - > Keys
 - > Constraints
 - > Triggers
 - > db.DP_Procedures
 - > db.DL_Procedures
 - > db.DM_Procedures

Results Messages

assembly_id	process_id
1	1
2	2
3	3
4	4
5	5
6	3
7	6
8	3
9	7
10	4
10	10

Line 1, Col 27 Spaces: 4 UTF-8 LF SQL MSOL 10 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db

Fifth:

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | . . . | HelloWorld.java | . . .
. . .
671     case "4":
672         System.out.println("Please enter a new Assembly ID:");
673     . . .
674 }
. . .
9) Retrieve the total cost incurred on an assembly-line;
10) Retrieve the cost of the department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve the cost of a certain burndate given date in a given department;
13) Retrieve the customer id (name, order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Insert a new row in "Paint_20";
16) Import: enter new customers from a data file;
20) Exit;
. . .
Please enter a new Assembly ID:
5
Please enter the Date ordered for the assembly:
11/12/20
Please enter the Assembly details:
. . .
Please enter the associated Customer Name:
. . .
Please enter the number of associated process IDs with this assembly
Please enter 0th process ID:
1
Please enter 1th process ID:
. . .
Please enter 2th process ID:
. . .
Please enter 3th process ID:
. . .
Dispatching the query 4(a)...
Done, 1 rows inserted in "Assembly" table.
Dispatching the query 4(b)...
Done, 1 rows inserted in "order1" table.
Dispatching the query 4(c)...
Done, 1 rows inserted in "passthru" table.
Dispatching the query 4(d)...
Done, 1 rows inserted in "passthru" table.
Dispatching the query 4(e)...
Done, 1 rows inserted in "passthru" table.
. . .
. . .
Done, 1 rows inserted in "passthru" table.

Please select one of the options below:

```

Results:

File Edit View Help

CONNECTIONS

SQL Query_1 - gaur0001-sql-server.database.windows.net-dba-4513-sql-db (gaur0001) - Azure Data Studio

Run Close Document Change Connection (s-dba-4513-sql-db) Export Enter SQLCMD

1. select * from Assembly;

Results Messages

	assembly_id	assembly_details	date_ordered
1	1	rgh	10102011
2	2	tyu	10152020
3	3	qwe	10102020
4	4	wer	11102013
5	5	ert	11112010

servers

- labor_time (int, null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- dbo.Process
 - dbo.job
 - dbo.Movie
 - dbo.Orders
 - dbo.assembly
 - Columns
 - name (PK, FK, varchar20), assembly_id (PK, FK, int, not null)
 - assembly_id (PK, FK, int, not null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.Paint.job
 - dbo.Paint.Process
- dbo.assembly
 - Columns
 - assembly_id (PK, int, not null)
 - paint_type (varchar20), null
 - painting_method (varchar20), null
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
- dbo.Paint.Through
 - Columns
 - Assembly_M (PK, FK, int, not null), Assembly_F (PK, FK, int, not null)
 - process_id (PK, FK, int, not null)
 - Keys
 - Constraints
 - Triggers

File Edit View Help

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Order

RESULTS

name	assembly_id
James	3
2	2
Richard	4
Smith	2
Steve	5

ADO

SQL SERVER BIG DATA CLUSTERS

Line 1, Col 21 Spaces: 4 UTF-8 LF SQL MSOLK 5 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

File Edit View Help

Welcome SQLQuery_3.sql - gaur00..uri0001 SQLQuery_1 - gaur00..uri0001

Run Disconnect Change Connection cs-dba-4513-sql-db Explain Execute TSQL

1 select * from Pass_Through

RESULTS

assembly_id	process_id
1	1
2	1
2	2
3	2
4	2
5	2
6	3
7	3
8	3
9	3
10	4
10	10
11	5
12	5
13	5
14	5

ADO

SQL SERVER BIG DATA CLUSTERS

Line 1, Col 27 Spaces: 4 UTF-8 LF SQL MSOLK 14 rows 00:00:08 gaur0001-sql-server.database.windows.net.cs-dba-4513-sql-db

Sixth:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
# | Package Explorer | 0 HelloWorld.java | 1 |
| 0 HelloWorld.java | 2 | |
| 3 case "4": | 4 System.out.println("Please enter a new Assembly ID:"); | 5 |
| 6 | 7 Problems = javadoc Declaration | 8 |
| 9 AzurSQLDBC [java Application] AzurSQLDBC.java | 10 openjdk hotspot jre full linux x86_64 16.0.1 v20210526-1205/rebir/java (Nov 22, 2021, 10:47:33 PM) | 11 |
| 12 Done. 1 rows inserted in "passthru" table. | 13 |
| 14 Dispatching the query 4(b)... | 15 |
| 16 Done. 1 rows inserted in "passthru" table. | 17 |
| 18 Please select one of the options below: | 19 |
| 19 1) Enter New Customer; | 20 |
| 20 2) Enter a new department; | 21 |
| 21 3) Enter a new Employee; | 22 |
| 22 4) Enter a new Assembly; | 23 |
| 23 5) Enter a new Account Number; | 24 |
| 24 6) Enter a new Job; | 25 |
| 25 7) Enter the Completion Date of a Job; | 26 |
| 26 8) Enter the Completion Date of an Order; | 27 |
| 27 9) Retrieve the total cost incurred on an assembly_id; | 28 |
| 28 10) Retrieve the total cost incurred on an order_id for a given day; | 29 |
| 29 11) Retrieve the process through which a given assembly_id has passed; | 30 |
| 30 12) Retrieve the jobs completed during given date in a given department; | 31 |
| 31 13) Retrieve the assembly_id of an order whose category is in a given range; | 32 |
| 32 14) Delete a job with a Job Number; | 33 |
| 33 15) Change the cut of a Paint Job; | 34 |
| 34 16) Insert customers from a data file; | 35 |
| 35 17) Export Customer Data; | 36 |
| 36 18) Exit! | 37 |
| 37 | 38 Please enter a new Assembly ID: | 39 |
| 39 | 40 Please enter the Date ordered for the assembly: | 41 |
| 41 | 42 Please enter the Assembly details: | 43 |
| 43 | 44 Please enter the associated Customer Name: | 45 |
| 45 | 46 Please enter the number of associated process IDs with this assembly | 47 |
| 47 | 48 Please enter nth process ID: | 49 |
| 49 | 50 Dispatching the query 4(a)... | 51 |
| 51 | 52 Done. 1 rows inserted in "Assembly" table. | 53 |
| 53 | 54 Dispatching the query 4(b)... | 55 |
| 55 | 56 Done. 1 rows inserted in "order" table. | 57 |
| 57 | 58 Dispatching the query 4(c)... | 59 |
| 59 | 60 Done. 1 rows inserted in "passthru" table. | 61 |
| 61 | 62 Please select one of the options below: | 63 |

```

Results:

assembly_id	assembly_details.date_ordered
1	19930201
2	19930202
3	19930203
4	19930204
5	19930205
6	19930206

Seventh:

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Search Project Run Window Help
# eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
1 package AzureSql;
2
3 import java.util.*;
4
5 public class HelloWorld {
6
7     public static void main(String[] args) {
8
9         case "4":
10            System.out.println("Please enter a new Assembly ID:");
11
12            AzureSQLJDBC.main((java Application) $url$eclipse$plugins$org.eclipse.jdt$openjdk.hotspot.jre.full.linux.x86_64_16.0.1$20210526-1205$rebin$java (Nov 22, 2021, 10:47:33 PM)
13            // 1. Enter a new Assembly:
14            // 2. Enter a new Account Number:
15            // 3. Enter the completion date of a Job:
16            // 4. Retrieve the total cost incurred on an assembly-Id:
17            // 5. Retrieve total labor within a Department for a given day:
18            // 6. Retrieve the jobs completed during a given date in a given department:
19            // 7. Enter the completion date of a Job:
20            // 8. Enter a new Assembly ID:
21            // 9. Retrieve the total cost incurred on an assembly-Id:
22            // 10. Retrieve total labor within a Department for a given day:
23            // 11. Retrieve the jobs completed during a given date in a given department:
24            // 12. Retrieve the jobs completed during given date in a given department:
25            // 13. Retrieve all customers with a job Number:
26            // 14. Delete all cut-jobs with a Job Number:
27            // 15. Change the color of a Paint Job:
28            // 16. Insert a new customer from a data file:
29            // 17. Export Customer Data:
30            // 20) Exit!
31
32            Please enter a new Assembly ID:
33
34            Please enter the Date ordered for the assembly:
35            yui
36            Please enter the Assembly details:
37            John
38            Please enter the associated Customer Name:
39            John
40            Please enter the number of associated process IDs with this assembly
41            3
42            Please enter 1th process ID:
43            Please enter 2th process ID:
44            Dispatching the query 4(a)...
45            Done. 1 rows inserted in "Assembly" table.
46            Dispatching the query 4(b)...
47            Done. 1 rows inserted in "order1" table.
48            Dispatching the query 4(b)...
49            Done. 1 rows inserted in "process" table.
50            Dispatching the query 4(b)...
51            Done. 1 rows inserted in "passthrough" table.
52            Dispatching the query 4(b)...
53            Done. 1 rows inserted in "passthrough" table.
54            Done. 1 rows inserted in "passthrough" table.
55
56            Please select one of the options below:

```

Results:

assembly_id	assembly_name	date_ordered
1	John	10/10/2011
2	yui	10/13/2010
3	que	10/03/2010
4	wer	11/10/2013
5	ert	11/11/2010
6	yui	7/7/2013
7	yui	23/03/2001

Eight:

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Project Run Window Help
# package AzureSql;
# import java.util.*;
# import org.json.*;
# import com.microsoft.sqlserver.jdbc.*;

public class AzureSQLDBC {
    public static void main(String[] args) {
        try {
            String assembly_id = args[0];
            switch (assembly_id) {
                case "1":
                    System.out.println("Please enter a new Assembly ID:");
                    break;
                case "2":
                    System.out.println("Please enter a new Customer Name:");
                    break;
                case "3":
                    System.out.println("Please enter a new Process:");
                    break;
                case "4":
                    System.out.println("Please enter a new Order Number:");
                    break;
                case "5":
                    System.out.println("Please enter a new Account Number:");
                    break;
                case "6":
                    System.out.println("Please enter a new Job:");
                    break;
                case "7":
                    System.out.println("Please enter the creation date of a Job:");
                    break;
                case "8":
                    System.out.println("Please enter a new Transaction Number:");
                    break;
                case "9":
                    System.out.println("Please enter the assembly id for an assembly-id");
                    break;
                case "10":
                    System.out.println("Please enter the total labor time within a given day:");
                    break;
                case "11":
                    System.out.println("Please enter the process through which a given assembly id has passed:");
                    break;
                case "12":
                    System.out.println("Please enter the customer name for a given department:");
                    break;
                case "13":
                    System.out.println("Please enter the customers (in name order) whose category is in a given range:");
                    break;
                case "14":
                    System.out.println("Please enter the cut-job id for a job number:");
                    break;
                case "15":
                    System.out.println("Please enter the cut-job id for a job number:");
                    break;
                case "16":
                    System.out.println("Import: enter new customers from a data file:");
                    break;
                case "17":
                    System.out.println("Import: enter Customer Data");
                    break;
                case "20":
                    System.out.println("Exit");
                    break;
                default:
                    System.out.println("Please enter a new Assembly ID:");
            }
            System.out.println("Please enter the Date ordered for the assembly:");
            String date_ordered = System.console().readLine();
            System.out.println("Please enter the Assembly details:");
            String assembly_details = System.console().readLine();
            System.out.println("Please enter the associated Customer Name:");
            String customer_name = System.console().readLine();
            System.out.println("Please enter the number of associated process IDs with this assembly");
            int process_ids = Integer.parseInt(System.console().readLine());
            System.out.println("Please enter 6th process ID:");
            int process_id_6 = Integer.parseInt(System.console().readLine());
            System.out.println("Please enter 10th process ID:");
            int process_id_10 = Integer.parseInt(System.console().readLine());
            System.out.println("Dispatching the query 4...");
            String query4 = "select * from Assembly";
            System.out.println("Done. 1 rows inserted in \"Assembly\" table.");
            System.out.println("Dispatching the query 4b...");
            String query4b = "select * from OrderDetail";
            System.out.println("Done. 1 rows inserted in \"OrderDetail\" table.");
            System.out.println("Dispatching the query 4c...");
            String query4c = "select * from PassThrough";
            System.out.println("Done. 1 rows inserted in \"PassThrough\" table.");
            System.out.println("Done. 1 rows inserted in \"PassThrough\" table.");
            System.out.println("Please select one of the options below:");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Results:

assembly_id	assembly_name	date_ordered
1	James	20100201
2	John	20100202
3	Richard	20100202
4	John	20100203
5	John	20100203
6	John	20100203
7	John	20100203
8	John	20100203

Ninth:

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Search Project Run Window Help
# package AzureSql;
# import java.util.Scanner;
# public class HelloWorld {
#     public static void main(String[] args) {
#         Scanner scanner = new Scanner(System.in);
#         int choice = scanner.nextInt();
#         switch (choice) {
#             case 1:
#                 System.out.println("Please enter a new Customer ID:");
#                 break;
#             case 2:
#                 System.out.println("Please enter a new Account ID:");
#                 break;
#             case 3:
#                 System.out.println("Please enter a new Process ID:");
#                 break;
#             case 4:
#                 System.out.println("Please enter a new Order ID:");
#                 break;
#             case 5:
#                 System.out.println("Please enter a new Account Number:");
#                 break;
#             case 6:
#                 System.out.println("Please enter a new Job ID:");
#                 break;
#             case 7:
#                 System.out.println("Please enter the completion date of a Job:");
#                 break;
#             case 8:
#                 System.out.println("Please enter a new Transaction Number:");
#                 break;
#             case 9:
#                 System.out.println("Please enter a new Assembly ID:");
#                 break;
#             case 10:
#                 System.out.println("Please enter the total labor time within a Department for a given day:");
#                 break;
#             case 11:
#                 System.out.println("Please enter the process through which a given assembly id has passed:");
#                 break;
#             case 12:
#                 System.out.println("Please enter the order number for a given assembly id and a given department:");
#                 break;
#             case 13:
#                 System.out.println("Please enter the customers (in name order) whose category is in a given range:");
#                 break;
#             case 14:
#                 System.out.println("Please enter the cut-off date of a Job ID:");
#                 break;
#             case 15:
#                 System.out.println("Please enter the cut-off date of a Job ID:");
#                 break;
#             case 16:
#                 System.out.println("Please enter the customer details from a data file:");
#                 break;
#             case 17:
#                 System.out.println("Please enter the customer data:");
#                 break;
#             case 18:
#                 System.out.println("Please enter the number of associated process IDs with this assembly");
#                 break;
#             case 19:
#                 System.out.println("Please enter the nth process ID:");
#                 break;
#             case 20:
#                 System.out.println("Please enter the 10th process ID:");
#                 break;
#             default:
#                 System.out.println("Please enter a valid choice.");
#         }
#     }
# }

```

Results:

assembly_id	assembly_details.date_ordered
1	James
2	Tyler
3	Mike
4	Wendy
5	Art
6	Yuri
7	Steve
8	Bob
9	Art

Tenth:

```

eclipse-workspace - AzureSql\src\AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | 0 HelloWorld.java | 1 |
71 case "4": System.out.println("Please enter a new Assembly ID:");
72 
73 // Problems = javadoc Declaration
74 
75 // [1] Enter a new Account Number:
76 // [2] Enter a new Assembly ID:
77 // [3] Enter the Completion date of a Job:
78 // [4] Enter the Date ordered for the assembly:
79 // [5] Enter the total cost incurred on an assembly-Id
80 // [6] Retrieve the total labor within a Department for a given day:
81 // [7] Retrieve the total number of jobs completed during a given date in a given department:
82 // [8] Retrieve the jobs completed during given date in a given department:
83 // [9] Retrieve the job completed during given date in a given department:
84 // [10] Retrieve the total cost of a job:
85 // [11] Retrieve the total cost of a job:
86 // [12] Retrieve the total cost of a job:
87 // [13] Retrieve the total cost of a job:
88 // [14] Delete all cut-jobs with a Job Number:
89 // [15] Change the color of a Paint Job:
90 // [16] Delete all customers from a data file:
91 // [17] Export Customer Data:
92 // [20] Exit!
93 
94 // Please enter a new Assembly ID:
95 // [2]
96 // Please enter the Date ordered for the assembly:
97 // [4]
98 // Please enter the Assembly details:
99 // [5]
100 // Please enter the associated Customer Name:
101 // Steve
102 // Please enter the number of associated process IDs with this assembly
103 // 3
104 // Please enter 1th process ID:
105 // Please enter 2th process ID:
106 // Dispatching the query 4(a)...
107 // Done. 1 rows inserted in "Assembly" table.
108 // Dispatching the query 4(b)...
109 // Done. 1 rows inserted in "order1" table.
110 // Dispatching the query 4(c)...
111 // Done. 1 rows inserted in "passthru" table.
112 // Dispatching the query 4(d)...
113 // Done. 1 rows inserted in "passthru" table.
114 // Dispatching the query 4(e)...
115 // Done. 1 rows inserted in "passthru" table.
116 // Done. 1 rows inserted in "passthru" table.
117 // Done. 1 rows inserted in "passthru" table.
118 
119 // Please select one of the options below:

```

Results:

assembly_id	assembly_detail	date_ordered
1	gft	10102011
2	tyu	10152020
3	qwe	10102020
4	wer	11102013
5	ert	11112010
6	yui	7072013
7	ytu	20051201
8	uor	7072020
9	ert	2002
10	iop	2003

name	assembly_id
James	3
James	6
John	1
John	7
Richard	4
Richard	9
Seith	2
Seith	8
Steve	5
Steve	10

assembly_id	process_id
16	7
17	7
18	7
19	1
20	8
21	9
22	9
23	10
24	10
25	10

Ten Queries of Type 5:

First:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is:

```

297     System.out.println("Executing Statement 9 if...:");
298     System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rows_inserted));
299 }
300
301 else {
302     // Information for Cut Process
303 }
304
305
306 // AzuresQlJDBC [java Application] Asnrl/eclipse/plugins/org.eclipse.jdt/openjdk.hotspot.jre.full.linux.x86_64_16.0.1/v20210526-1205/rebir/java (Nov 22, 2021, 10:47:33 PM)
307
308 1) Enter a new department;
309 2) Enter a new department;
310 3) Enter a new Process ;
311 4) Enter a new Process ;
312 5) Enter a new Account Number;
313 6) Enter a new Job;
314 7) Enter the completion date of a Job;
315 8) Enter a new Transaction Number;
316 9) Enter a new department based on an assembly_id;
317 10) Retrieve total labor time within a Department for a given day;
318 11) Retrieve the process through which a given assembly was passed;
319 12) Retrieve the department account for a given department;
320 13) Retrieve the customers (in name order) whose category is in a given range;
321 14) Enter a new department number;
322 15) Change the color of a Paint Job;
323 16) Import: enter new customers from a data file;
324 17) Enter Customer Data;
325 18) Enter Customer Data;
326 19) Exit!
327
328 Please enter a new Account number
329
330 Please enter the date of account establishment:
331 2003
332 Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:
333
334 Please enter the department number associated with the account:
335 1
336 Connecting to the database...
337 Dispatching the query 5(1)... .
338 Done. 1 rows inserted in "account" table.
339 Dispatching the query 5(2)... .
340 Done. 1 rows inserted in "departmentaccount" table .
341 Dispatching the query 5(3)... .
342 Done. 1 rows inserted in "record_dept_cost" table .
343
344 Please select one of the options below:
345 1) Enter new customer;
346 2) Enter new department;
347 3) Enter a new department;

```

Results:

The screenshot shows the Microsoft SQL Server Data Studio interface with the title "SQLQuery_1 - gaur001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur001) - Azure Data Studio".

Results pane:

- Query 1: `select * from Account;`

account_id	account
1	12
2	2003

- Query 2: `select * from Dept_Account;`

account_id	sup_cost
1	12
2	0

- Query 3: `select * from Record_Dept_Cost;`

account_id	dept_id
1	12
2	11

Second:

```

public class HelloWorld {
    public static void main(String[] args) {
        int rowInserted = 0;
        String query = "INSERT INTO record_assembly_cost (assembly_id, account_id, sup_cost) VALUES (?, ?, ?);";
        try (Connection conn = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=master;integratedSecurity=true");
             PreparedStatement pstmt = conn.prepareStatement(query)) {
            for (int i = 0; i < 1000; i++) {
                pstmt.setInt(1, i);
                pstmt.setInt(2, i);
                pstmt.setDouble(3, 100.0);
                pstmt.executeUpdate();
                rowInserted++;
                System.out.println("Executing Statement " + i + "...");
                System.out.println(String.format("Done. %d rows inserted in \\\"record_assembly_cost\\\" table.", rowInserted));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

The screenshot shows the Eclipse IDE interface with the code editor open. The code implements a JDBC connection to a SQL Server database and performs a bulk insert operation into the 'record_assembly_cost' table. It also includes logic to handle user input for various database operations like inserting into 'account', selecting from 'Assembly_Account', and inserting into 'assembly_by_account'.

Results:

account_id	sup_cost
34	0

The screenshot shows the Microsoft Data Studio interface with a query results pane. The query selected account_id and sup_cost from record_assembly_cost where account_id = 34. The result set contains a single row with account_id 34 and sup_cost 0.

Third:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is:

```

291     System.out.println("Executing Statement 9 if...:");
292     System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cast\\ table.", rows_inserted));
293
294     }
295
296     }
297
298     else {
299
300         // Information for Cut_Process
301     }
302 }

```

The output window shows the execution of the code:

```

AzurSQLDBC [java Application] /usr/lib/eclipse/plugins/org.eclipse.justjavajdk/hotspot/pre.full/linux.x86_64_16.0.1/v20210526-1205/prebuilt/java (Nov 22, 2021, 10:47:33 PM)
1) Enter a new customer;
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new Account ;
5) Enter a new Account Number;
6) Enter a new Job ;
7) Enter a new location date of a Job;
8) Enter a new Transaction Number;
9) Enter a new Customer Data;
10) Retrieve total labor time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve all the customers whose date of birth is a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a job number;
15) Insert a new customer;
16) Import: enter new customers from a data file;
17) Update Customer Data;
20) Exit;

Please enter a new Account number
45
Please enter the date of account establishment:
11112003
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:
1
Please enter the associated process ID:
Connecting to the database...
Dispatching the query $1...
Done. 1 row inserted in "account" table.
Dispatching the query $1...
Done. 1 row inserted in "processaccount" table.
Dispatching the query $1...
Done. 1 row inserted in "record_process_cost" table.

Please select one of the options below:
1) Enter new Customer;
1) Enter new Customer...

```

Results:

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The results pane displays the following data:

account_id	process_id
12	2003
34	12112014
45	11112003

Fourth:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the `HelloWorld.java` file containing Java code for a `Cut_Process` class.
- Terminal:** Shows a terminal window with the following text:


```
282 System.out.println("Executing Statement 9 if...");  
283 System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rows_inserted));  
284  
285     }  
286  
287 }  
288 else {  
289 // Information for Cut_Process  
290  
291 // 1) Enter new customer:  
292 // 2) Enter a new department;  
293 // 3) Enter a new Process ;  
294 // 4) Enter a new Account Number;  
295 // 5) Enter a new Job;  
296 // 6) Enter a new Transaction Number;  
297 // 7) Enter a new Department;  
298 // 8) Enter a new Assembly id;  
299 // 9) Enter a new Customer id;  
300 // 10) Retrieve total labor time within a Department for a given day;  
301 // 11) Retrieve the process through which a given assembly-id has passed;  
302 // 12) Retrieve all the customers whose date of birth is a given department;  
303 // 13) Retrieve the customers (in name order) whose category is in a given range;  
304 // 14) Delete all cut-jobs with a job number;  
305 // 15) Insert a new record in Payroll table;  
306 // 16) Import: enter new customers from a data file;  
307 // 17) Import: customer Data  
308 }  
  
Please enter a new Account number  
or  
Please enter the date of account establishment:  
12232014  
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:  
2  
Please enter the department number associated with the account:  
12  
Connecting to the database...  
Dispatching the query S1a...  
Done. 1 row inserted in "account" table.  
Dispatching the query S1b...  
Done. 1 row inserted in "dept_account" table .  
Dispatching the query S1c...  
Done. 1 row inserted in "record_dept_cost" table .  
Please select one of the options below:  
1) Enter new Customer;
```

Results:

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

- Table View:** A table titled "Results" showing data from the `dbo.Account` table. The columns are `account_id` and `sup_cost`. The data is as follows:

account_id	sup_cost
1	12
2	34
3	45
4	66

Fifth:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | src | AzureSQLDBC.java | 291
+-- AzureSQL
+-- HelloWorld
src | AzureSQLDBC.java | 292
293     System.out.println("Executing Statement 9/1/..."); 294
295     System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rows_inserted));
296
297     }
298
299     else { 300
300         // Information for Cut Process
301     }
302 }
303
304 public static void main(String[] args) {
305     try {
306         System.out.println("Executing Statement 9/1/..."); 307
308         System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rows_inserted));
309
310     } catch (Exception e) {
311         e.printStackTrace();
312     }
313 }
314
315 }
316
317 
```

AzureSQLDBC [java Application] Asnrl/recipse/plugins/org.eclipse.jdt/openjdk.hotspot.jre.full.linux.x86_64_16.0.1/v20210526-1205/rebir/java (Nov 22, 2021, 10:47:33 PM)

3) Enter a new Process :
4) Enter a new Customer :
5) Enter a new Account Number:
6) Enter a new Process ID:
7) Enter the completion date of a Job:
8) Enter a new Transaction Number:
9) Enter a new Assembly ID:
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the jobs completed during a date range in a given department:
12) Retrieve the jobs completed during a date in a given department:
13) Retrieve the customers (in name order) whose category is in a given range:
14) Retrieve the assembly ID for a given number:
15) Change the color of a Paint Job:
16) Import: enter new customers from a data file:
17) Import: Customer Data
20) Exit!

Please enter a new Account number
0
Please enter the date of account establishment:
10/22/2002
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:
2
Please enter the associated Assembly Id:
Connecting to the database...
Dispatching the query 5/1/...
Done. 1 rows inserted in "account" table.
Dispatching the query 5/1/...
Executing Statement 5/1/...
Done. 1 rows inserted in "assemblyaccount" table.
Dispatching the query 5/1/...
Executing Statement 5/1/...
Done. 1 rows inserted in "record_assembly_cost" table.

Please select one of the options below:
1) Enter new Customer:
2) Enter new Account Number.

Results:

File Edit View Help

CONNECTIONS | Welcome | SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513.sql-db (gaur0001) - Azure Data Studio

gaur0001-sql-server database window

Tables | dbo.Account | Results | Messages

account_id account_date

1	34	2003
2	89	12/11/2014

account_id sup_cust

1	34	0
2	89	0

account_id assembly_id

1	34	1
2	89	2

Le 8 Col 10 Spans: 4 UTF-8 LF SQL MODE: 9-Nov-2020 gaur0001-sql-server.database.windows.net.cs-dsa-4513.sql-db

Sixth:

```

public class HelloWorld {
    public static void main(String[] args) {
        int rows_inserted;
        String query = "SELECT * FROM record_assembly_cost";
        System.out.println("Executing Statement 9 if...:");
        System.out.println(String.format("Done. %d rows inserted in \"%s\" table.", rows_inserted));
        System.out.println("else {");
        System.out.println("    // information for Cut Process");
    }
}

```

The code continues with a series of numbered steps for managing assemblies and processes:

- 1) Enter a new Account Number:
- 2) Enter the Completion Date of a Job:
- 3) Enter the Job Number:
- 4) Retrieve the total cost incurred on a specific date:
- 5) Retrieve the total cost incurred in a department on a given day:
- 6) Retrieve the process through which a given assembly has passed:
- 7) Retrieve the jobs completed during given date in a given department:
- 8) Insert the total cost of all jobs whose category is in a given range:
- 9) Delete all cut-jobs with a Job Number:
- 10) Change the color of a Paint Job:
- 11) Insert or update customers from a data file:
- 12) Export Customer Data
- 13) Exit

Below the numbered steps, there are several prompts for user input:

- Please enter a new Account number
- Please enter the date of account establishment
- Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 for Process Acc
- Please enter the associated process ID:
- Connecting to the database...
- Dispatching the query 51a...
- Done. 1 rows inserted in "account" table.
- Done. 1 rows inserted in "processaccount" table.
- Dispatching the query 51b...
- Done. 1 rows inserted in "processcost" table..
- Please select one of the options below:
- 1) Enter new customer;
- 2) Enter a new department;
- 3) Enter a new Process;
- 4) Enter a new Assembly;
- 5) Enter a new Account Number;
- 6) Enter a new Job;

Results:

account_id	process_id
45	1
101	2

Seventh:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the `HelloWorld.java` file containing Java code for a `Cut_Process` class.
- Terminal:** Shows a terminal window with the following text:


```
287 System.out.println("Executing Statement 8 if...");  
288 System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rows_inserted));  
289  
290 }  
291  
292 else {  
293 // Information for Cut_Process.  
294  
295 // 1) Enter a new customer;  
296 // 2) Enter a new department;  
297 // 3) Enter a new Process ;  
298 // 4) Enter a new Account Number;  
299 // 5) Enter a new Account Number;  
300 // 6) Enter a new Job ;  
301 // 7) Enter a new duration date of a Job;  
302 // 8) Enter a new Transaction Number;  
303 // 9) Enter a new Department Account ;  
304 // 10) Retrieve total labor time within a Department for a given day;  
305 // 11) Retrieve the process through which a given assembly-id has passed;  
306 // 12) Retrieve all the customers whose date of birth is a given department;  
307 // 13) Retrieve the customers (in name order) whose category is in a given range;  
308 // 14) Delete all cut-jobs with a job number;  
309 // 15) Insert a new record in Payroll ;  
310 // 16) Import: enter new customers from a data file;  
311 // 17) Import: Customer Data  
312 // 20) Exit  
  
Please enter a new Account number  
47  
Please enter the date of account establishment:  
4801988  
Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:  
13  
Please enter the department number associated with the account:  
13  
Connecting to the database...  
Dispatching the query S1a...  
Done. 1 row inserted in "account" table.  
Dispatching the query S1b...  
Done. 1 row inserted in "dept_account" table .  
Dispatching the query S1c...  
Done. 1 row inserted in "record_dept_cost" table .  
Please select one of the options below:  
1) Enter new Customer;  
1) Enter a new Department;
```

Results:

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

- SQL Editor:** Shows a query result grid titled "Results" with the following data:

account_id	account
12	2003
34	12112014
45	11112003
47	4001988
66	12232014
89	10232002
101	7072000

Eight:

```

public class HelloWorld {
    public static void main(String[] args) {
        int rowInserted = 0;
        while(true) {
            System.out.println("Executing Statement 5/1/...:");
            System.out.println(String.format("Done. %d rows inserted in \\"record_assembly_cost\\ table.", rowInserted));
            System.out.println("1) Enter a new Process :");
            System.out.println("2) Enter a new Account Number:");
            System.out.println("3) Enter a new Customer Data:");
            System.out.println("4) Enter the completion date of a Job:");
            System.out.println("5) Enter a new Transaction Number:");
            System.out.println("6) Enter a new Assembly ID:");
            System.out.println("7) Enter total labor time within a Department for a given day:");
            System.out.println("8) Retrieve the jobs completed during a date range in a given department:");
            System.out.println("9) Retrieve the customers (in name order) whose category is in a given range:");
            System.out.println("10) Retrieve the color of a Paint Job:");
            System.out.println("11) Change the color of a Paint Job:");
            System.out.println("12) Import: enter new customers from a data file:");
            System.out.println("13) Insert Customer Data:");
            System.out.println("14) Exit:");
            System.out.println("Please enter a new Account number");
            String accountNumber = scanner.nextLine();
            System.out.println("Please enter the date of account establishment:");
            String accountEstablishmentDate = scanner.nextLine();
            System.out.println("Please enter the account type: 1 For Dept Acc, 2 For Assembly Acc & 3 For Process Acc:");
            String accountType = scanner.nextLine();
            System.out.println("Please enter the associated Assembly Id:");
            String assemblyId = scanner.nextLine();
            System.out.println("Connecting to the database...");
            DatabaseConnection dbConn = DatabaseConnection.getConnection();
            dbConn.createStatement();
            System.out.println("Done. 1 rows inserted in \"account\" table.");
            Dispatching the query 5/1/...
            Executing Statement 5/1/...
            Done. 1 rows inserted in "assemblyaccount" table.
            Dispatching the query 5/1/...
            Executing Statement 5/1/...
            Done. 1 rows inserted in "record_assembly_cost" table.
            Please select one of the options below:
            1) Enter new Customer;
            14) Insert Customer Data...
        }
    }
}

```

Results:

account_id	assembly_id
12	2003
34	12112014
45	11112003
47	40011986
66	12232014
89	10232002
91	1012020
101	7072000

Ninth:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the `HelloWorld.java` file containing Java code for a `Cut_Process` class.
- Log View:** Shows the execution log for the application, listing various steps and their results, such as inserting rows into tables like `record_assembly_cost` and `process`.
- Console View:** Displays a series of numbered steps from 1 to 20, likely representing a process flow or a set of instructions.
- Output View:** Shows the output of the application's execution, including prompts for user input like "Please enter a new Account number" and "Please enter the date of account establishment".

Results:

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

- SQL Editor:** Contains a query window with the following SQL code:


```
1 select * from Account;
2 select * from Process_Account;
3 select * from Record_Process_Cost;
```
- Results View:** Displays the results of the query, showing three columns of data:

account_id	account	process_id
12	2000	
34	12112014	
45	11112003	
47	40010986	
66	12232014	
86	7012021	
89	10232002	
91	10122020	
101	7012000	
45	0	
86	3	
101	0	

Tenth:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the `HelloWorld.java` file containing Java code for a JDBC application.
- Terminal:** Shows a series of SQL queries being executed against a database. The queries include:
 - Creating a table `Customer` with columns `customer_id`, `customer_name`, and `customer_address`.
 - Creating a table `Dept_Account` with columns `account_id`, `dept_id`, and `accnt_dept_cost`.
 - Inserting data into `Customer` and `Dept_Account`.
 - Selecting data from `Customer` and `Dept_Account`.
 - Joining `Customer` and `Dept_Account` to show department names.
 - Creating a view `Customer_Dept_Cost` combining `Customer` and `Dept_Account`.
 - Creating a stored procedure `Customer_Dept_Cost` that returns the view.
 - Calling the stored procedure `Customer_Dept_Cost` to display results.

Results:

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

- Connections:** A connection to `gaur0001-sql-server.database.windows.net.cs-dsa-4513.sql-db (gaur0001)` is selected.
- Tables:** The `Customer` table is expanded, showing columns `customer_id`, `customer_name`, and `customer_address`. It contains 10 rows of data.
- Tables:** The `Dept_Account` table is expanded, showing columns `account_id`, `dept_id`, and `accnt_dept_cost`. It contains 10 rows of data.
- Results:** A query result grid displays the joined data from both tables. The columns are `account_id`, `customer_name`, `customer_address`, `dept_id`, and `accnt_dept_cost`. The data shows 10 rows of customer information with their corresponding department names and costs.

Ten Queries for Type 6:

One

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". In the center, there is a code editor window titled "HelloWorld.java" containing Java code. The code includes a switch statement with options for creating a new customer, process, account, job, or exiting. It also includes logic for retrieving assembly details, processes, and customers, as well as inserting data into the "Job" and "Assign" tables. Below the code editor, the terminal window displays the execution of the application, showing user prompts and the resulting database insertions.

```

    ...
    case '6':
        break;
    case '7':
        sc.out.println("Please enter the Job Number:");
        final int jobno = sc.nextInt();
        sc.nextLine();
        sc.nextLine();
        System.out.println("Please enter the date of job commenced:");
        ...
        Please enter the Job Number:
        87072021
        Please enter the Process ID assigned to this job:
        1
        Please enter the Assembly ID assigned to the process for this job:
        2
        Dispatching the query 6(a)...
        Done. 1 rows inserted in "Job" table..
        Dispatching the query 6(b)...
        Done. 1 rows inserted in "Assign" table..
    ...

```

Results:

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513.sql-db (gaur0001) - Azure Data Studio". The left sidebar shows the database structure with tables like "Jobs", "Processes", "Assign", and "Assembly". The main area shows the results of two SQL queries. The first query, "select * from Job", returns one row with job_id 1, date_comenced '2022-01-01', date_completed null, labor_time 00:00:00, and job_type null. The second query, "select * from Assign", returns one row with job_id 1, assembly_id 2, and process_id 1.

job_id	date_comenced	date_completed	labor_time	job_type
1	2022-01-01		00:00:00	null

job_id	assembly_id	process_id
1	2	1

Two:

```

eclipse-workspace - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | ...
HelloWorld.java | ...
1188     final String csthmax = sc.nextLine();
1189     System.out.println("Please enter the output file name (please don't give the extension)");
1190     String filename = sc.nextLine();
1191     System.out.println("Connecting to the database..."); 
1192     // Get the database connection, create statement and execute it right away, as no user input need be collected
1193     try {
1194         Connection connection = DriverManager.getConnection(url);
1195         System.out.println("Dispatching the query..."); 
1196         try {
1197             final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1198             statement2.setString1(csthmax);
1199             statement2.setString2(csthmax);
1200             System.out.println("Executing the query..."); 
1201             Statement statement1 = connection.createStatement();
1202             ResultSet rs1 = statement1.executeQuery();
1203             // Unpack the tuples returned by the database and print them out to the user
1204             try {
1205                 PrintWriter writer = new PrintWriter(new File(filename + ".csv"));
1206             }
1207         }
1208     }
1209     catch (Exception e) {
1210         e.printStackTrace();
1211     }
1212     finally {
1213         System.out.println("Done. " + rows + " rows inserted in " + "Job" + " table..");
1214         System.out.println("Dispatching the query 6..."); 
1215         Statement statement2 = connection.createStatement();
1216         ResultSet rs2 = statement2.executeQuery();
1217         System.out.println("Done. " + rows + " rows inserted in " + "Assign" + " table..");
1218     }
1219     System.out.println("Please select one of the options below:");
1220 }

```

Problems > [details] Declaration []

Autodesk Live Application: /usr/lib/eclipse/plugins/org.eclipse.jdt.core/openjdk.hotspot/pre.full/linux/x86_64/16.0.1.v20210528-1205/rebir/java (Nov 23, 2021, 8:52:59 AM)

* Enter a new AssemblyID:

4) Enter a new Job ID:

7) Enter the Completion date of a Job:

8) Enter the Job Number:

9) Retrieve the total cost incurred on an assembly-ID

10) Retrieve total labor time within a Department for a given day.

11) Retrieve the department ID for which a job has passed.

12) Retrieve the jobs completed during given date in a given department:

13) Delete all jobs from a department in order

14) Delete all cut-jobs with a Job Number

15) Change the color of a Painter

16) Delete all customers from a data file

17) Export Customer Data

20) Exit!

Please enter the Job Number:

Please enter the date of job commenced:

Please enter the Process ID assigned to this job:

Please enter the Assembly ID assigned to the process for this job:

Dispatching the query 6...

Done. 1 rows inserted in "Job" table..

Dispatching the query 6...;

Done. 1 rows inserted in "Assign" table..

Please select one of the options below:

Results:

File Edit View Help

CONNECTIONS

gaur0001-sql-server-database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

gaur0001-sql-server-database.windows.net.cs-dsa-4513-sql-db

1. select * from Job;

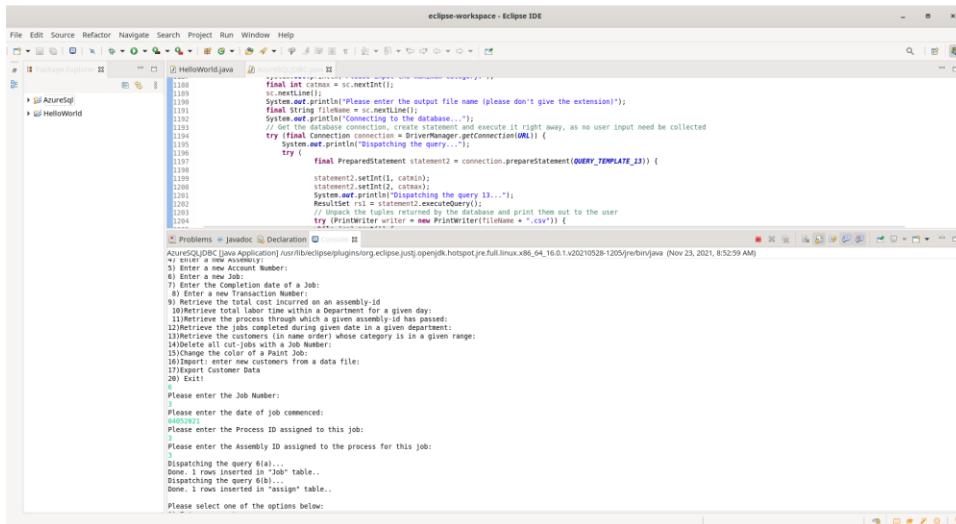
2. select * from Assign;

Results Messages

job_id	date_comenced	date_completed	labor_time	job_type
1	NULL	20220221	NULL	NULL
2	NULL	20220220	NULL	NULL

job_id	assembly_id	process_id
1	1	2
2	2	1

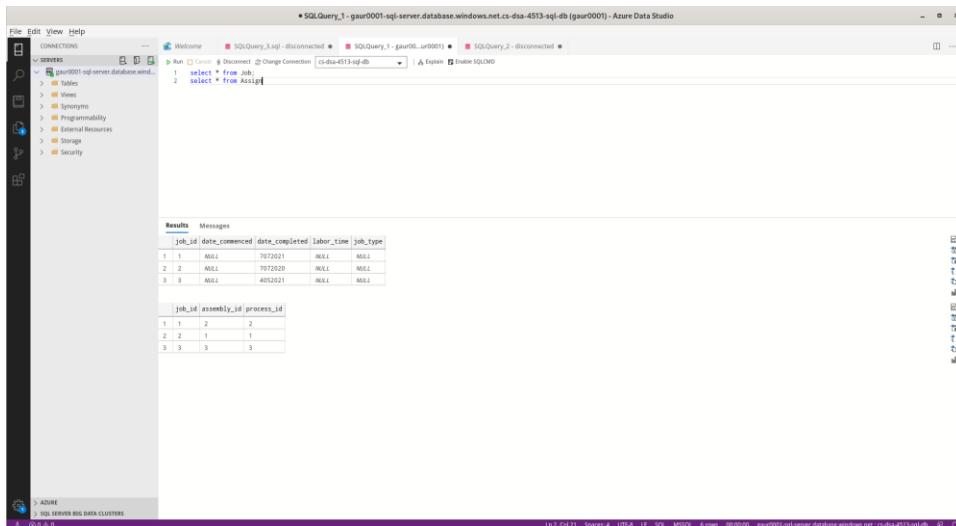
Three



The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the `HelloWorld.java` file containing Java code for connecting to a database and executing a query.
- Console:** Shows command-line arguments being passed to the application, including the database URL, port, and connection properties.
- Output:** Displays the execution results, including the output of the `System.out.println` statements and the results of the database query.

Results:



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

- Connections:** Shows the connection to the `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`.
- Tables:** Shows the `Job` and `Assign` tables.
- Results:** Displays the results of the executed SQL queries:

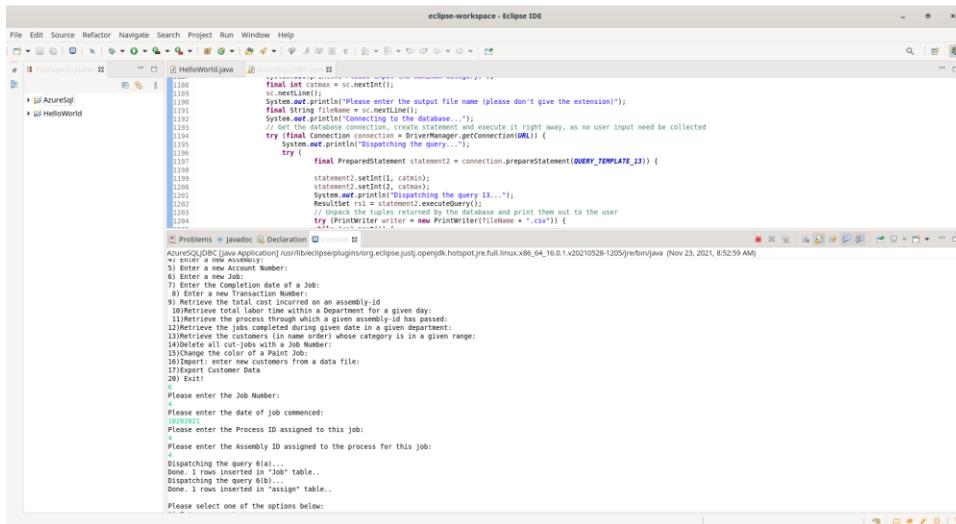
 - Query 1 (from Job):

job_id	date_committed	date_completed	labor_time	job_type
1	NULL	7072021	NULL	NULL
2	NULL	7072020	NULL	NULL
3	NULL	4032020	NULL	NULL

 - Query 2 (from Assign):

job_id	assembly_id	process_id
1	1	2
2	2	1
3	3	3

Fourth:



```

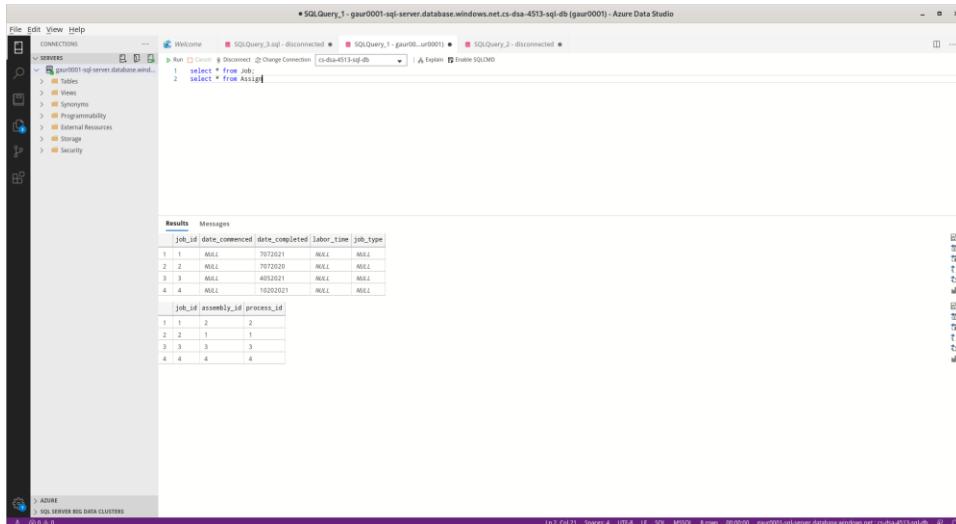
eclipse-workspace - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
HelloWorld.java [2]
1188     final String csthml = sc.next();
1189     final String nextline;
1190     System.out.println("Please enter the output file name (please don't give the extension):");
1191     System.out.println("Connecting to the database..."); 
1192     // Get the database connection, create statement and execute it right away, as no user input need be collected
1193     try {
1194         Connection connection = DriverManager.getConnection(url);
1195         System.out.println("Dispatching the query..."); 
1196         try {
1197             final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1198             statement2.setString1(csthml);
1199             statement2.setString2(csthml);
1200             System.out.println("The query is being executed now..."); 
1201             ResultSet rs1 = statement2.executeQuery();
1202             // Unpack the tuples returned by the database and print them out to the user
1203             try (PrintWriter writer = new PrintWriter(new File(filename + ".csv"))){ 
1204                 while(rs1.next()){
1205                     writer.println(rs1.getString("job_id") + "," + rs1.getString("assembly_id") + "," + rs1.getString("process_id"));
1206                 }
1207             }
1208         } catch (SQLException e) {
1209             e.printStackTrace();
1210         }
1211     } catch (SQLException e) {
1212         e.printStackTrace();
1213     }
1214     System.out.println("The job has been completed successfully!");
1215 }
1216
1217 public static void main(String[] args) {
1218     new HelloWorld();
1219 }
1220 }

Problems > Javadoc > Declaration > Documentation
AUTOMATICALLY (Give Application) /usr/lib/eclipse/plugins/org.eclipse.jdt.core/jdtopenjdk.hotspot/pre.full/linux,x86_64,16.0.1.v20210528-1205/rebir/java (Nov 23, 2021, 8:52:59 AM)
*+ Create a new Account:
    1) Enter a new Account Number:
    2) Enter the Completion date of a Job:
    3) Enter the Assembly ID assigned to this job:
    4) Enter the Process ID assigned to this job:
    5) Enter the Job Number:
    6) Enter the Date of job commenced:
    7) Enter the completion date of a job:
    8) Enter the assembly ID assigned to this job:
    9) Retrieve the total cost incurred on an assembly-ID
    10) Retrieve total labor time within a Department for a given day:
    11) Retrieve all the jobs completed during a given date in a given department:
    12) Retrieve the jobs completed during given date in a given department:
    13) Delete all the jobs with a Job Number in a given range:
    14) Delete all cut-jobs with a Job Number:
    15) Change the color of a Paint Job:
    16) Insert customers from a data file:
    17) Export Customer Data:
    18) Exit!
    ...
    Please enter the Job Number:
    ...
    Please enter the date of job commenced:
    ...
    Please enter the Process ID assigned to this job:
    Please enter the Assembly ID assigned to the process for this job:
    ...
    Dispatching the query...
Done, 4 rows inserted in "Job" table..
Dispatching the query 6...
Done, 4 rows inserted in "Assign" table..
Please select one of the options below:

```

Results:



SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

Results

job_id	date_committed	date_completed	labor_time	job_type
1	NULL	20220221	NULL	NULL
2	NULL	20220220	NULL	NULL
3	NULL	20220221	NULL	NULL
4	NULL	20220221	NULL	NULL

job_id	assembly_id	process_id
1	1	2
2	2	1
3	3	3
4	4	4

Fifth

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". In the center, there is a code editor window titled "HelloWorld.java" containing Java code. Below the code editor is a terminal window showing the execution of the Java application. The terminal output includes several numbered steps (1 through 18) and a final message "Please select one of the options below:" followed by a prompt for a job number.

```

1188         final String csthml = sc.next();
1189         System.out.println("Please enter the output file name (please don't give the extension)");
1190         System.out.println("Connecting to the database..."); 
1191         // Get the database connection, create statement and execute it right away, as no user input need be collected
1192         try {
1193             Connection connection = DriverManager.getConnection(url);
1194             System.out.println("Dispatching the query..."); 
1195             try {
1196                 final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1197                 statement2.setString1(csthml);
1198                 statement2.setString2(csthml);
1199                 System.out.println("Executing the query..."); 
1200                 Statement statement1 = connection.createStatement();
1201                 ResultSet rs1 = statement1.executeQuery("Select * from Job");
1202                 System.out.println("Unpacking the tuples returned by the database and print them out to the user");
1203                 try (PrintWriter writer = new PrintWriter(new File(filename + ".csv"))){ 
1204                     while(rs1.next()){
1205                         writer.println(rs1.getString("Job_ID") + "," + rs1.getString("Assembly_ID") + "," + rs1.getString("Process_ID"));
1206                     }
1207                 }
1208             } catch (SQLException e) {
1209                 e.printStackTrace();
1210             }
1211         } catch (SQLException e) {
1212             e.printStackTrace();
1213         }
1214         System.out.println("Please enter the Job Number:");
1215         String jobNumber = scanner.nextLine();
1216         System.out.println("Please enter the date of job commenced:");
1217         String dateCommenced = scanner.nextLine();
1218         System.out.println("Please enter the Process ID assigned to this job:");
1219         String processID = scanner.nextLine();
1220         System.out.println("Please enter the Assembly ID assigned to the process for this job:");
1221         String assemblyID = scanner.nextLine();
1222         System.out.println("Dispatching the query...");
1223         String sql = "Insert into Job values (" + jobNumber + ", " + assemblyID + ", " + processID + ", " + dateCommenced + ")";
1224         Statement statement = connection.createStatement();
1225         int rowsAffected = statement.executeUpdate(sql);
1226         System.out.println(rowsAffected + " rows inserted in " + "Job" + " table..");
1227         System.out.println("Dispatching the query...");
1228         sql = "Insert into Assign values (" + jobNumber + ", " + assemblyID + ", " + processID + ", " + dateCommenced + ")";
1229         Statement statement2 = connection.createStatement();
1230         rowsAffected = statement2.executeUpdate(sql);
1231         System.out.println(rowsAffected + " rows inserted in " + "Assign" + " table..");
1232         System.out.println("Please select one of the options below:");
1233     }
    
```

Results

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". On the left, the object browser shows a connection to "gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db". The main area displays the results of a SQL query. The first result set shows data from the "Job" table:

job_id	date_commenced	date_completed	labor_time	job_type
1	NULL	7072021	NULL	NULL
2	NULL	7072020	NULL	NULL
3	NULL	4032021	NULL	NULL
4	NULL	10202021	NULL	NULL
5	NULL	10212021	NULL	NULL

The second result set shows data from the "Assign" table:

job_id	assembly_id	process_id
1	1	2
2	1	1
3	2	3
4	4	4
5	6	5

Sixth

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". On the left, the "Package Explorer" shows two projects: "AzureSQL" and "HelloWorld". The "HelloWorld" project has a "src" folder containing a "HelloWorld.java" file. The code in "HelloWorld.java" is as follows:

```

1 package com.hello;
2
3 import java.sql.*;
4
5 public class HelloWorld {
6     public static void main(String[] args) {
7         final String ctmname = args[0];
8         final String extname = args[1];
9         System.out.println("Please enter the output file name (please don't give the extension)");
10        System.out.println("Connecting to the database..."); 
11        // Get the database connection, create statement and execute it right away, as no user input need be collected
12        try {
13            Connection connection = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test");
14            System.out.println("Dispatching the query..."); 
15            final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
16            statement2.setString1(ctmname);
17            statement2.setString2(ctmname);
18            System.out.println("Executing the query..."); 
19            Statement statement1 = connection.createStatement();
20            ResultSet rs1 = statement1.executeQuery();
21            System.out.println("Unpacking the tuples returned by the database and print them out to the user");
22            try {
23                PrintWriter writer = new PrintWriter(new File(ctmname + ".csv"));
24                while (rs1.next()) {
25                    writer.println(rs1.getString("job_id") + "," + rs1.getString("assembly_id") + "," + rs1.getString("process_id"));
26                }
27                writer.close();
28            } catch (Exception e) {
29                e.printStackTrace();
30            }
31        } catch (Exception e) {
32            e.printStackTrace();
33        }
34    }
35 }

```

The "Problems" view shows a warning: "No source code is available for type: org.eclipse.jdt.core.IJavaElement". The "Console" tab shows the command "cd /opt/eclipse/plugins/org.eclipse.jdt.core/jdtCompilerEngine/bin" and the output of the Java application's execution.

Result:

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The left sidebar shows the "CONNECTIONS" tree with "gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db" selected. The main area displays the results of two SQL queries:

```

1 select * from Job;
2 select * from Asig;

```

Results

job_id	date_created	date_completed	labor_time	job_type
1	NULL	7072021	NULL	NULL
2	NULL	7072021	NULL	NULL
3	NULL	4032021	NULL	NULL
4	NULL	10202021	NULL	NULL
5	NULL	10212021	NULL	NULL
6	NULL	10212021	NULL	NULL

job_id	assembly_id	process_id
1	2	2
2	2	1
3	3	3
4	4	4
5	5	5
6	6	5

Seventh:

eclipse-workspace - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | Javadoc | Declaration | Problems | Javadoc | Declaration | Problems |
HelloWorld.java | 21
1188         final int cmax = sc.nextInt();
1189         sc.nextLine();
1190         System.out.println("Please enter the output file name (please don't give the extension)");
1191         System.out.println("Connecting to the database..."); 
1192         // Get the database connection, create statement and execute it right away, as no user input need be collected
1193         try {
1194             Connection connection = DriverManager.getConnection(url);
1195             System.out.println("Dispatching the query..."); 
1196             try {
1197                 final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1198                 statement2.setInt(1, cmax);
1199                 statement2.setInt(2, cmax);
1200                 System.out.println("Executing the query..."); 
1201                 Statement statement1 = connection.createStatement();
1202                 ResultSet rs1 = statement1.executeQuery();
1203                 // Unpack the tuples returned by the database and print them out to the user
1204                 try (PrintWriter writer = new PrintWriter(new File(filename + ".csv"))){ 
1205                     while(rs1.next()){
1206                         writer.println(rs1.getString("job_id") + "," + rs1.getString("assembly_id") + "," + rs1.getString("process_id"));
1207                     }
1208                 }
1209             } catch (SQLException e) {
1210                 e.printStackTrace();
1211             }
1212         } catch (SQLException e) {
1213             e.printStackTrace();
1214         }
1215     }
1216     System.out.println("Please enter the Job Number:");
1217     jobNumber = sc.nextInt();
1218     sc.nextLine();
1219     System.out.println("Please enter the date of job commenced:");
1220     dateCommenced = sc.next();
1221     sc.nextLine();
1222     System.out.println("Please enter the Process ID assigned to this job:");
1223     processID = sc.nextInt();
1224     sc.nextLine();
1225     System.out.println("Please enter the Assembly ID assigned to the process for this job:");
1226     assemblyID = sc.nextInt();
1227     sc.nextLine();
1228     System.out.println("Dispatching the query...");
1229     Done: if (rs1.next())
1230         System.out.println("1 rows inserted in " + "Job" + " table..");
1231     System.out.println("Dispatching the query 6...");
1232     Done: if (rs1.next())
1233         System.out.println("1 rows inserted in " + "Assign" + " table..");
1234     System.out.println("Please select one of the options below:");
1235 
```

Results:

SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

File Edit View Help

CONNECTIONS

- gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - disconnected
- SQLQuery_2 - disconnected
- SQLQuery_3 - disconnected

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

RESULTS

Results Message

job_id	date_commenced	date_completed	labor_time	job_type
1	NULL	7/3/2021	NULL	NULL
2	NULL	7/3/2020	NULL	NULL
3	NULL	4/3/2021	NULL	NULL
4	NULL	7/3/2020	NULL	NULL
5	NULL	7/11/2021	NULL	NULL
6	NULL	10/21/2021	NULL	NULL
7	NULL	10/28/2021	NULL	NULL

job_id	assembly_id	process_id
1	1	2
2	2	1
3	3	3
4	4	4
5	5	5
6	6	5
7	7	7

SQL SERVER BIG DATA CLUSTERS

Line 2, Col 21 | Spaces 4 | UTF-8 | LF | SQL | MySQL | 14 rows | 00:00:00 | gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db | P | O

Eighth:

eclipse-workspace - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | Javadoc | Declaration | Problems | Javadoc | Declaration | Problems |
HelloWorld.java | 21
1188     final String ctnamex = sc.next();
1189     sc.nextLine();
1190     System.out.println("Please enter the output file name (please don't give the extension)");
1191     System.out.println("Connecting to the database..."); 
1192     // Get the database connection, create statement and execute it right away, as no user input need be collected
1193     try {
1194         Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/test");
1195         System.out.println("Dispatching the query..."); 
1196         final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1197         statement2.setString1(ctnamex);
1198         statement2.setString2(ctnamex);
1199         System.out.println("Executing the query..."); 
1200         Statement statement1 = connection.createStatement();
1201         ResultSet rs1 = statement1.executeQuery("SELECT * FROM job");
1202         System.out.println("Unpacking the tuples returned by the database and print them out to the user");
1203         try (PrintWriter writer = new PrintWriter(new File(ctnamex + ".csv"))){ 
1204             while(rs1.next()){
1205                 writer.println(rs1.getString("job_id") + "," + rs1.getString("assembly_id") + "," + rs1.getString("process_id"));
1206             }
1207         }
1208         System.out.println("Disposing the connection");
1209         connection.close();
1210     } catch (SQLException e) {
1211         e.printStackTrace();
1212     }
1213     System.out.println("Job completed successfully");
1214 }
1215
1216 public static void main(String[] args) {
1217     new HelloWorld();
1218 }
1219
1220 
```

Problems | Javadoc | Declaration | Problems | Javadoc | Declaration | Problems |

AutodeskJava Application | Azul/eclipse/plugins/org.eclipse.jdt.core/jdtopenjdk.hotspot/pre.full/linux,x86_64,16.0.1.v20210528-1205/rebir/java [Nov 23, 2021, 9:12:49 AM]

* Enter a new AssemblyId:
 45
 45
 7) Enter the Completion date of a Job:
 9) Retrieve the total cost incurred on an assembly-Id
 10) Retrieve total labor time within a Department for a given day.
 11) Retrieve all the jobs assigned to a department whose id has passed:
 12) Retrieve the jobs completed during given date in a given department:
 13) Delete all cut-jobs with a Job Number
 15) Change the color of a Paint Job:
 16) Insert new customers from a data file:
 17) Export Customer Data
 20) Exit!

Please enter the Job Number:
 45
 Please enter the date of job commenced:
 10/12/2021
 Please enter the Process ID assigned to this job:
 Please enter the Assembly ID assigned to the process for this job:
 Dispatching the query...
 Done. 1 rows inserted in "Job" table..
 Dispatching the query 6/6...
 Done. 4 rows inserted in "Assign" table..
 Please select one of the options below:

Results:

SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

File Edit View Help

CONNECTIONS

- gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - disconnected
- SQLQuery_1 - gaur00..ur0001 - disconnected
- SQLQuery_2 - disconnected

RESULTS

Results Message

job_id	date_comenced	date_completed	labor_time	job_type
1	NULL	7/3/2021	NULL	NULL
2	NULL	7/3/2020	NULL	NULL
3	NULL	4/3/2021	NULL	NULL
4	NULL	7/3/2021	NULL	NULL
5	NULL	1/11/2021	NULL	NULL
6	NULL	10/21/2021	NULL	NULL
7	NULL	10/28/2021	NULL	NULL
8	NULL	1/10/2021	NULL	NULL

job_id	assembly_id	process_id
1	1	2
2	2	1
3	3	3
4	4	4
5	5	5
6	6	5
7	7	2
8	8	8

ADVISOR

SQL SERVER BIG DATA CLUSTERS

File 2, Col 21 | Spaces 4 | UTF-8 | SQL | MySQL | 14 rows | 00:00:00 | gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db | P | O

Ninth:

eclipse-workspace - Eclipse IDE

```

File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
HelloWorld.java 21
1188         final String cstrmax = sc.nextInt();
1189         sc.nextLine();
1190         System.out.print("Please enter the output file name (please don't give the extension):");
1191         System.out.println("Connecting to the database..."); 
1192         // Get the database connection, create statement and execute it right away, as no user input need be collected
1193         try {
1194             Connection connection = DriverManager.getConnection(url);
1195             System.out.println("Dispatching the query..."); 
1196             try {
1197                 final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1198                 statement2.setString(1, cstrmax);
1199                 statement2.setString(2, cstrmax);
1200                 System.out.println("Executing the query..."); 
1201                 Statement statement1 = connection.createStatement();
1202                 ResultSet rs1 = statement1.executeQuery();
1203                 // Unpack the tuples returned by the database and print them out to the user
1204                 try (PrintWriter writer = new PrintWriter(new File(filename + ".csv"))){ 
1205                     while(rs1.next()){
1206                         writer.println(rs1.getString("job_id") + "," + rs1.getString("assembly_id") + "," + rs1.getString("process_id"));
1207                     }
1208                 }
1209             } catch (SQLException e) {
1210                 e.printStackTrace();
1211             }
1212         } catch (SQLException e) {
1213             e.printStackTrace();
1214         }
1215     }
1216     public static void main(String[] args) {
1217         new HelloWorld().run();
1218     }
1219 }

```

Problems * (No errors or warnings)

AUTORUN/Java Application: /usr/lib/eclipse/plugins/org.eclipse.jdt.core/jre/lib/linux-x86_64/16.0.1.v20210526-1205/jre/bin/java (Nov 23, 2021, 9:12:49 AM)

* Create a new Assembly:

1) Enter the Job Number:

2) Enter the Completion date of a Job:

3) Enter the Assembly ID assigned to this job:

4) Enter the Process ID assigned to this job:

5) Please enter the Job Number:

6) Please enter the date of job commenced:

7) Please enter the Process ID assigned to this job:

8) Please enter the Assembly ID assigned to the process for this job:

9) Dispatching the query... Done, 1 rows inserted in "Job" table..

10) Retrieving total labor time within a Department for a given day...

11) Retrieving all the jobs completed during a given date in a given department...

12) Retrieving the jobs completed during given date in a given department:

13) Deleting all the jobs in a given department...

14) Deleting all cut-jobs with a Job Number...

15) Change the color of a Paint Job...

16) Deleting all customers from a data file...

17) Export Customer Data...

18) Exit!

Please enter the Job Number:

Please enter the date of job commenced:

Please enter the Process ID assigned to this job:

Please enter the Assembly ID assigned to the process for this job:

9) Dispatching the query... Done, 1 rows inserted in "Job" table..

Dispatching the query 6(s)..

Done, 1 rows inserted in "Assign" table..

Please select one of the options below:

Results:

File Edit View Help

CONNECTIONS

SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio

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

1 select * from Job;

2 select * from Assign;

Results

job_id	date_committed	date_completed	labor_time	job_type
1	NULL	7/3/2021	NULL	NULL
2	NULL	7/3/2020	NULL	NULL
3	NULL	4/3/2021	NULL	NULL
4	NULL	7/3/2021	NULL	NULL
5	NULL	7/11/2021	NULL	NULL
6	NULL	10/21/2021	NULL	NULL
7	NULL	10/28/2021	NULL	NULL
8	NULL	11/17/2021	NULL	NULL
9	NULL	11/21/2021	NULL	NULL

job_id	assembly_id	process_id
1	1	2
2	2	1
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9

File Edit View Help

SQL SERVER BIG DATA CLUSTERS

File Edit View Help

File Edit View Help

Tenth:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - Eclipse IDE". In the center, there is a code editor window for "HelloWorld.java" containing Java code. Below the code editor is a terminal window showing the execution of the Java application. The terminal output includes a list of 17 numbered options for interacting with a database, followed by a prompt for a job number and date, and then a series of "Done." messages indicating successful database operations.

```

1188         final String csthmax = sc.next();
1189         sc.nextLine();
1190         System.out.println("Please enter the output file name (please don't give the extension)");
1191         String csthname;
1192         System.out.println("Connecting to the database..."); 
1193         // Get the database connection, create statement and execute it right away, as no user input need be collected
1194         try {
1195             Connection connection = DriverManager.getConnection(url);
1196             System.out.println("Dispatching the query..."); 
1197             try {
1198                 final PreparedStatement statement2 = connection.prepareStatement(QUERY_TEMPLATE_1);
1199                 statement2.setString(1, csthname);
1200                 statement2.setString(2, csthmax);
1201                 System.out.println("Executing the query..."); 
1202                 Statement statement1 = connection.createStatement();
1203                 ResultSet rs1 = statement1.executeQuery();
1204                 // Unpack the tuples returned by the database and print them out to the user
1205                 try {
1206                     PrintWriter writer = new PrintWriter(new File(filename + ".csv"));
1207                 }
1208             }
1209         } catch (Exception e) {
1210             e.printStackTrace();
1211         }
1212     }
1213     System.out.println("Please enter the Job Number:");
1214     String jobNumber;
1215     jobNumber = scanner.nextLine();
1216     System.out.println("Please enter the date of job commenced:");
1217     String dateCommenced;
1218     dateCommenced = scanner.nextLine();
1219     System.out.println("Please enter the Process ID assigned to this job:");
1220     String processID;
1221     processID = scanner.nextLine();
1222     System.out.println("Please enter the Assembly ID assigned to the process for this job:");
1223     String assemblyID;
1224     assemblyID = scanner.nextLine();
1225     System.out.println("Dispatching the query...");
1226     Done. 1 rows inserted in "Job" table..
1227     Dispatching the query 6...
1228     Done. 4 rows inserted in "Assign" table..
1229     Please select one of the options below:
    
```

Results:

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". It displays two SQL queries and their results. The first query is "select * from Job;" and the second is "select * from Assign;". The results show data for the Job table (10 rows) and the Assign table (10 rows), respectively.

job_id	message	date_commenced	date_completed	labor_time	job_type
1	NULL	19700101	NULL	NULL	NULL
2	NULL	19700201	NULL	NULL	NULL
3	NULL	19700301	NULL	NULL	NULL
4	NULL	19700401	NULL	NULL	NULL
5	NULL	19700501	NULL	NULL	NULL
6	NULL	19700601	NULL	NULL	NULL
7	NULL	19700701	NULL	NULL	NULL
8	NULL	19700801	NULL	NULL	NULL
9	NULL	19700901	NULL	NULL	NULL
10	NULL	19701001	NULL	NULL	NULL

job_id	assembly_id	process_id
1	2	2
2	1	1
3	3	3
4	4	4
5	5	5
6	6	5
7	7	7
8	8	8
9	9	9
10	10	10

Ten Queries for Type 7:

First

```

    final int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted in \\'Paint_Job\\' table..", rows_inserted));
}

3) Enter a new Process :
4) Enter a new Assembly:
5) Enter a new Department:
6) Enter a new Job:
7) Enter the Completion Date of a Job:
8) Enter the Job Number:
9) Retrieve the total cost incurred on an assembly-id given department-id for a specific date.
10) Retrieve all the jobs completed during a given day.
11) Retrieve the process through which a given assembly-id has passed.
12) Retrieve the jobs completed during given date in a given department.
13) Retrieve all the jobs completed in a department whose category is in a given range.
14) Delete all cut-jobs with a Job Number:
15) Import: enter new customers from a file "Paint Job".
16) Import: enter new customers from a data file:
17) Export Customer Data
18) Exit

Please enter the Job Number:
Please enter the date the job completed:
Please enter the kind of Job 1. for cut Job, 2 for Paint Job and 3 for Fit Job
Please enter the type of Machine used:
knife
Please enter the amount of time required in hours:
3
Please enter the labor time required in hours:
3
Please enter the material used for the job:
cardboard
Connecting to the database...
Dispatching the query 1...
Done. 1 rows inserted in "Job" table..
Dispatching the query 7...
Done. 1 rows inserted in "cutjob" table..

```

Results:

job_id	date_commenced	date_completed	labor_time	job_type
1	7/7/2021	7/10/2021	2	1
2	7/7/2021	NULL	NULL	NULL
3	7/8/2021	NULL	NULL	NULL
4	7/8/2021	NULL	NULL	NULL
5	7/11/2021	NULL	NULL	NULL
6	7/12/2021	NULL	NULL	NULL
7	7/12/2021	NULL	NULL	NULL
8	7/13/2021	NULL	NULL	NULL
9	7/13/2021	NULL	NULL	NULL
10	7/17/2021	NULL	NULL	NULL

job_id	machine_type	machine_time	material_used	labor_time
1	knife	2	cardboard	2

Second

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows two projects: AzureSql and HelloWorld.
- Code Editor:** Displays Java code for a class named `HelloWorld.java`. The code includes a method that prints a message and a loop that asks for user input to perform various tasks. A list of 17 tasks is provided at the bottom of the code editor.
- Task List:**
 - Enter a new Assembly;
 - Enter a new Account Number;
 - Enter the Completion date of a Job;
 - Enter a new Transaction Number;
 - Retrieve all the customers carried on an assembly_id;
 - Metrics total labor time within a Department for a given day;
 - Retrieve the jobs completed during given date in a given department;
 - Retrieve the customers (in name order) whose category is in a given range;
 - Change the color of a Paint Job;
 - Import new customers from a data file;
 - Export customer data;
 - Exit!

Results:

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

- File Bar:** File, Edit, View, Help.
- Connections:** Shows a connection to "gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)".
- Tables:** Shows the "PaintJob" table.
- Query Editor:** Contains the following SQL queries:


```
1 select * from Job;
2 select * from PaintJob;
```
- Results Grid:** Displays the results of the second query, showing 10 rows of data from the "PaintJob" table. The columns are: job_id, date_committed, date_completed, labor_time, job_type.

Third

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows packages AzureSql and HelloWorld.
- Code Editor:** Displays Java code for `HelloWorld.java`. The code includes a main method that inserts data into a `Paint_Job` table and prints the number of rows inserted. It also contains a series of numbered options for interacting with the database.
- Status Bar:** Shows the path `eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE`, the date `(Nov 23, 2021, 9:53:52 AM)`, and the message `System.out.println(String.format("Done.%d rows inserted in \\"Paint_Job\\" table..", row_inserted));`.

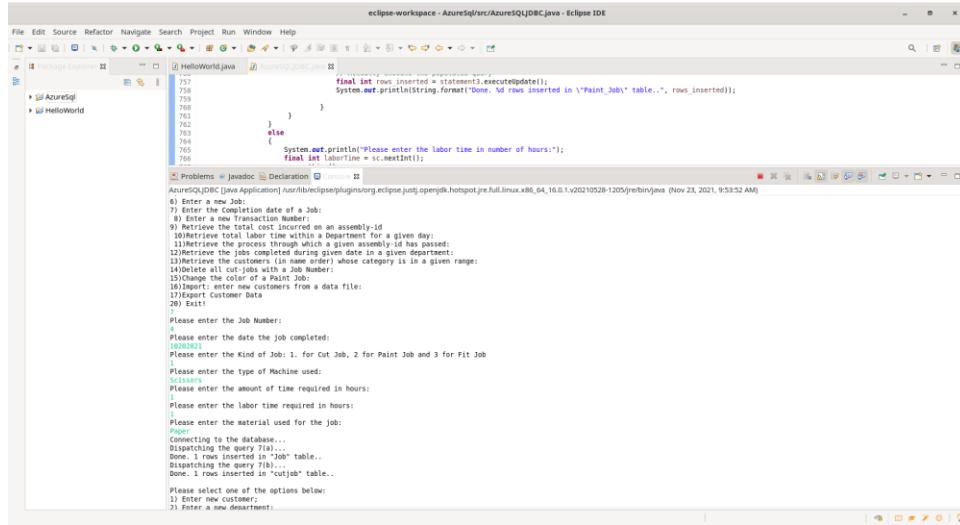
Results:

The screenshot shows the SSMS interface with the following details:

- File Bar:** File, Edit, View, Help.
- Connections:** Shows a connection to `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`.
- Object Explorer:** Shows the database structure, including tables like `dbo.Job`, `dbo.Cut_Job`, and `dbo.Fit_Job`.
- Results Grid:** Displays the results of a query against the `Fit_Job` table. The columns are `job_id`, `date_committed`, `date_completed`, `labor_time`, and `job_type`. The data is as follows:

job_id	date_committed	date_completed	labor_time	job_type
1	2021-07-01	2021-07-02	2	1
2	2021-07-02	2021-07-03	1	2
3	2021-07-03	2021-07-04	2	3
4	2021-07-04	2021-07-05	4	NULL
5	2021-07-05	2021-07-06	4	NULL
6	2021-07-06	2021-07-07	4	NULL
7	2021-07-07	2021-07-08	4	NULL
8	2021-07-08	2021-07-09	4	NULL
9	2021-07-09	2021-07-10	4	NULL
10	2021-07-10	2021-07-11	4	NULL

Fourth



```

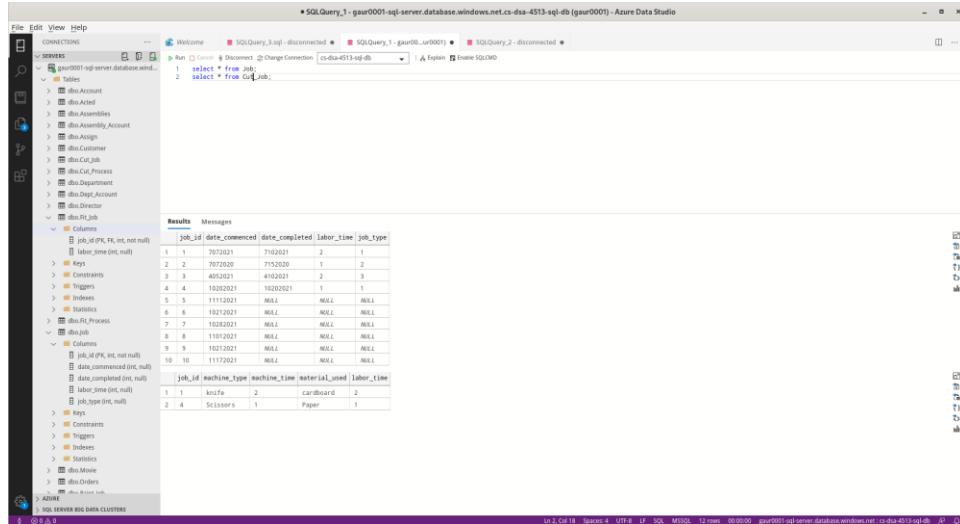
    757         final int rows_inserted = statement.executeUpdate();
    758         System.out.println(String.format("Done. %d rows inserted in \'%s\' table..", rows_inserted));
    759     }
    760     }
    761     else
    762     {
    763         System.out.println("Please enter the labor time in number of hours:");
    764         float labor_time = scanner.nextFloat();
    765         String query = "update Cut_Job set labor_time = " + labor_time;
    766     }
    767 }
    768 }

    769     System.out.println("Please enter the Job Number:");
    770     String jobNumber = scanner.nextLine();
    771     System.out.println("Please enter the date the job completed:");
    772     Date dateCompleted = scanner.nextDate();
    773     System.out.println("Please enter the kind of Job: 1, for Cut Job, 2 for Paint Job and 3 for Fit Job");
    774     int jobType = scanner.nextInt();
    775     System.out.println("Please enter the type of Machine used:");
    776     String machineType = scanner.nextLine();
    777     System.out.println("Please enter the amount of time required in hours:");
    778     float laborTime = scanner.nextFloat();
    779     System.out.println("Please enter the material used for the job:");
    780     String materialUsed = scanner.nextLine();
    781     scanner.nextLine();
    782     System.out.println("Connecting to the database...");
    783     Dispatching the query 7(a)...
    784     Done. 1 row inserted in "Cut_Job" table..
    785     Dispatching the query 7(b)...
    786     Done. 1 row inserted in "CutJob" table..
    787 }

    788     System.out.println("Please select one of the options below:");
    789     int option = scanner.nextInt();
    790     switch(option)
    791     {
    792         case 1:
    793             System.out.println("Enter new customer details:");
    794             scanner.nextLine();
    795             break;
    796         case 2:
    797             System.out.println("Enter new department details:");
    798             scanner.nextLine();
    799             break;
    800         case 3:
    801             System.out.println("Enter new job details:");
    802             scanner.nextLine();
    803             break;
    804         case 4:
    805             System.out.println("Enter new assembly details:");
    806             scanner.nextLine();
    807             break;
    808         case 5:
    809             System.out.println("Enter new employee details:");
    810             scanner.nextLine();
    811             break;
    812         case 6:
    813             System.out.println("Enter new machine details:");
    814             scanner.nextLine();
    815             break;
    816         case 7:
    817             System.out.println("Enter new department details:");
    818             scanner.nextLine();
    819             break;
    820         case 8:
    821             System.out.println("Enter new customer details:");
    822             scanner.nextLine();
    823             break;
    824         case 9:
    825             System.out.println("Enter new department details:");
    826             scanner.nextLine();
    827             break;
    828         case 10:
    829             System.out.println("Enter new job details:");
    830             scanner.nextLine();
    831             break;
    832         case 11:
    833             System.out.println("Enter new assembly details:");
    834             scanner.nextLine();
    835             break;
    836         case 12:
    837             System.out.println("Enter new employee details:");
    838             scanner.nextLine();
    839             break;
    840         case 13:
    841             System.out.println("Enter new machine details:");
    842             scanner.nextLine();
    843             break;
    844         case 14:
    845             System.out.println("Enter new department details:");
    846             scanner.nextLine();
    847             break;
    848         case 15:
    849             System.out.println("Enter new customer details:");
    850             scanner.nextLine();
    851             break;
    852         case 16:
    853             System.out.println("Enter new department details:");
    854             scanner.nextLine();
    855             break;
    856         case 17:
    857             System.out.println("Enter new job details:");
    858             scanner.nextLine();
    859             break;
    860         case 18:
    861             System.out.println("Enter new assembly details:");
    862             scanner.nextLine();
    863             break;
    864         case 19:
    865             System.out.println("Enter new employee details:");
    866             scanner.nextLine();
    867             break;
    868         case 20:
    869             System.out.println("Enter new machine details:");
    870             scanner.nextLine();
    871             break;
    872     }
    873 }
    874 }

```

Results:



job_id	date commenced	date completed	labor_time	job_type
1	2021-01-01	2021-01-01	2	1
2	2021-01-02	2021-01-02	1	2
3	2021-01-03	2021-01-03	2	3
4	2021-01-04	2021-01-04	1	1
5	2021-01-05	2021-01-05	1	2
6	2021-01-06	2021-01-06	1	3
7	2021-01-07	2021-01-07	1	2
8	2021-01-08	2021-01-08	1	3
9	2021-01-09	2021-01-09	1	2
10	2021-01-10	2021-01-10	1	3

job_id	machine_type	machine_time	material_used	labor_time
1	Knife	2	cardboard	2
2	Scissors	1	Paper	1

Fifth:

```

    final int jobTime = scanner.nextInt();
    System.out.println("Please enter the labor time in number of hours:");
    final int laborTime = scanner.nextInt();

    if (jobType == 1) {
        System.out.println("Please enter the date the job completed:");
        Date dateCompleted = scanner.nextDate();
        String formattedDate = dateCompleted.toString();
        System.out.println("Please enter the kind of Job: 1 for Cut Job, 2 for Paint Job and 3 for Fit Job");
        int jobKind = scanner.nextInt();
        System.out.println("Please enter the color of the paint used:");
        String jobColor = scanner.next();
        System.out.println("Please enter the volume of paint used:");
        double jobVolume = scanner.nextDouble();
        System.out.println("Please enter the labor time in number of hours:");
        double jobLaborTime = scanner.nextDouble();
        System.out.println("Connecting to the database...");
        Dispatching the query 7(1)...  

        Done. 1 row inserted in "Paint_Job" table.  

        Dispatching the query 7(1)...  

        Done. 1 row inserted in "Paint_Job" table...
    }
}
    
```

Results:

job_id	date_committed	date_completed	labor_time	job_type
1	2021-01-01	2021-01-01	2	1
2	2021-01-01	2021-01-01	1	2
3	2021-01-01	2021-01-01	2	3
4	2021-01-01	2021-01-01	1	1
5	2021-01-01	2021-01-01	1	2
6	2021-01-01	2021-01-01	NULL	NULL
7	2021-01-01	2021-01-01	NULL	NULL
8	2021-01-01	2021-01-01	NULL	NULL
9	2021-01-01	2021-01-01	NULL	NULL
10	2021-01-01	2021-01-01	NULL	NULL

Sixth:

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows packages AzureSql and HelloWorld.
- Code Editor:** Displays Java code for `HelloWorld.java`. The code includes a main method that prints "Hello World" and inserts rows into a `Paint_Job` table. It also contains a switch statement for user input regarding labor time.
- Console:** Shows the output of the Java application and a command-line interface for a database job. The command-line interface includes options for entering new customers, departments, and assembly numbers, as well as various queries related to jobs and departments.

Results:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface with the following details:

- File Bar:** File, Edit, View, Help.
- Connections:** Shows a connection to `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`.
- Object Explorer:** Shows the database structure, including tables like `dbo.Job`, `dbo.Cut_Job`, and `dbo.Fit_Job`.
- SQL Editor:** Contains the following SQL queries:


```
1 select * from Job;
2 select * from Fit_Job;
```
- Results Grid:** Displays the results of the second query, showing data for the `Fit_Job` table. The columns are `job_id`, `date_committed`, `date_completed`, `labor_time`, and `job_type`. The data is as follows:

job_id	date_committed	date_completed	labor_time	job_type
1	2021-01-01	2021-01-01	2	1
2	2021-01-01	2021-01-01	1	2
3	2021-01-01	2021-01-01	2	3
4	2021-01-01	2021-01-01	1	1
5	2021-01-01	2021-01-01	1	2
6	2021-01-01	2021-01-01	1	3
7	2021-01-01	2021-01-01	NULL	NULL
8	2021-01-01	2021-01-01	NULL	NULL
9	2021-01-01	2021-01-01	NULL	NULL
10	2021-01-01	2021-01-01	NULL	NULL

Seventh:

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows packages AzureSql and HelloWorld.
- Code Editor:** Displays Java code for `HelloWorld.java`. The code includes imports for `java.sql.*`, `java.util.*`, and `java.io.*`. It contains a main method with logic to insert data into a `Paint_Job` table and interact with a user for job information.
- Console:** Shows the output of the Java code execution, including SQL statements being sent to the database and their results.
- Status Bar:** Shows the path `eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE`.

Results:

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

- File Bar:** File, Edit, View, Help.
- Connections:** Shows a connection to `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`.
- Query Editor:** Contains two queries:
 - `select * from Job;`
 - `select * from Cut_Job;`
- Results Grid:** Displays the results of the two queries. The first query returns 10 rows of data for the `Job` table, and the second query returns 10 rows of data for the `Cut_Job` table.
- Object Explorer:** Shows the database schema, including tables like `Job`, `Cut_Job`, and `Machine`.
- Status Bar:** Shows the path `SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio`.

Eight:

```

    final int jobTime = scanner.nextInt();
    System.out.println("Please enter the labor time in number of hours:");
    final int userEnteredTime = scanner.nextInt();

    // Please enter the Job Number:
    // Please enter the date the job completed:
    // Please enter the kind of Job: 1. for Cut Job, 2 for Paint Job and 3 for Fit Job
    // Please enter the color of the paint used:
    // Please enter the volume of paint used:
    // Please enter the labor time in number of hours:
    // Connecting to the database...
    // Dispatching the query 7(1)...
    // Done. 1 row inserted in "Paint_Job" table.
    // Dispatching the query 7(1)...
    // Done. 1 row inserted in "Paint_Job" table.
    // Dispatching the query 7(1)...
    // Done. 1 row inserted in "Paint_Job" table.
    // Please select one of the options below:
    // 1) Enter new customer;
    // 2) Enter a new department;
  
```

Results:

job_id	date_committed	date_completed	labor_time	job_type
1	2021-01-01	2021-01-01	2	1
2	2021-01-01	2021-01-01	1	2
3	2021-01-01	2021-01-01	2	3
4	2021-01-01	2021-01-01	1	1
5	2021-01-01	2021-01-01	1	2
6	2021-01-01	2021-01-01	1	3
7	2021-01-01	2021-01-01	2	1
8	2021-01-01	2021-01-01	1	2
9	2021-01-01	2021-01-01	NULL	NULL
10	2021-01-01	2021-01-01	NULL	NULL

job_id	color	volume	labor_time
1	red	1	1
2	blue	2	1
3	green	4	1

Ninth:

The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows packages AzureSql and HelloWorld.
- Code Editor:** Displays Java code for `HelloWorld.java`:


```

757         final int rows_inserted = statement.executeUpdate();
758         System.out.println(String.format("Done.. %d rows inserted in \'%s\' table..", rows_inserted));
759     }
760     }
761     else
762     {
763         System.out.println("Please enter the labor time in number of hours:");
764     }
765     final int laborTime = scanner.nextInt();
766 }
```
- Console Output:**

```

eclipse-workspace - AzureSql\src\AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | HelloWord.java | ...
757         final int rows_inserted = statement.executeUpdate();
758         System.out.println(String.format("Done.. %d rows inserted in \'%s\' table..", rows_inserted));
759     }
760     }
761     else
762     {
763         System.out.println("Please enter the labor time in number of hours:");
764     }
765     final int laborTime = scanner.nextInt();
766 }

Problems in Javadoc Declaration
[1] AnseriSQLDBC [Java Application] AnseriSQLDBC\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205\jre\bin\java (Nov 23, 2021, 9:53:52 AM)
Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
3) Enter a new Process;
4) Enter a new Assembly;
5) Enter a new Category Number;
6) Enter a new Job;
7) Enter a new Employee;
8) Enter a new Transaction Number;
9) Enter a new Transaction date of a Job;
10) Enter a new Transaction time within a Department for a given day;
11) Retrieve the process through which a given assembly-id has passed;
12) Retrieve all the customers whose category date in a given department;
13) Retrieve the customers (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Import: enter new customers from a file;
16) Import: enter new customers from a data file;
17) Import: Customer Data;
20) Exit;

Please enter the Job Number:
Please enter the date the job completed:
10232021
Please enter the kind of Job: 1. for Cut Job, 2 for Paint Job and 3 for Fit Job
_
Please enter the labor time in number of hours:
Connecting to the database...
Dispatching the query 71(0)... Done. 1 row inserted in "cut_job" table.
Dispatching the query 71(0)... Done. 1 row inserted in "paint_job" table.
Dispatching the query 71(0)... Done. 1 row inserted in "fit_job" table.

Please select one of the options below:
1) Enter new customer;
2) Enter a new department;
```

Results:

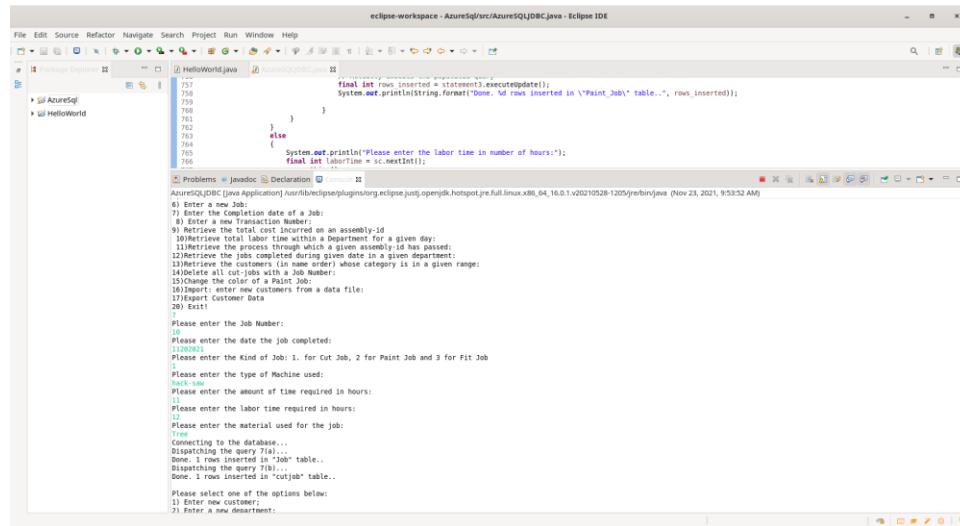
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface with the following details:

- File Bar:** File, Edit, View, Help.
- Object Explorer:** Shows the database structure for `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`, including tables like `dbo.Customer`, `dbo.Assembly`, `dbo.Cut_Job`, `dbo.Fit_Job`, `dbo.Paint_Job`, and `dbo.Process`.
- Query Editor:** Contains the following SQL query:


```
select * from Fit_Job;
```
- Results Grid:** Displays the results of the query:

job_id	Message	date_committed	date_completed	labor_time	job_type
1		7072021	7102021	2	1
2		7072020	7152020	1	2
3		4022021	41022021	2	3
4		1022021	1032021	1	1
5		11122021	11222021	1	2
6		10212021	10222021	1	3
7		10232021	10252021	2	1
8		11012021	11032021	1	2
9		10212021	10232021	3	3
10		11172021	NULL	NULL	NULL

Tenth:

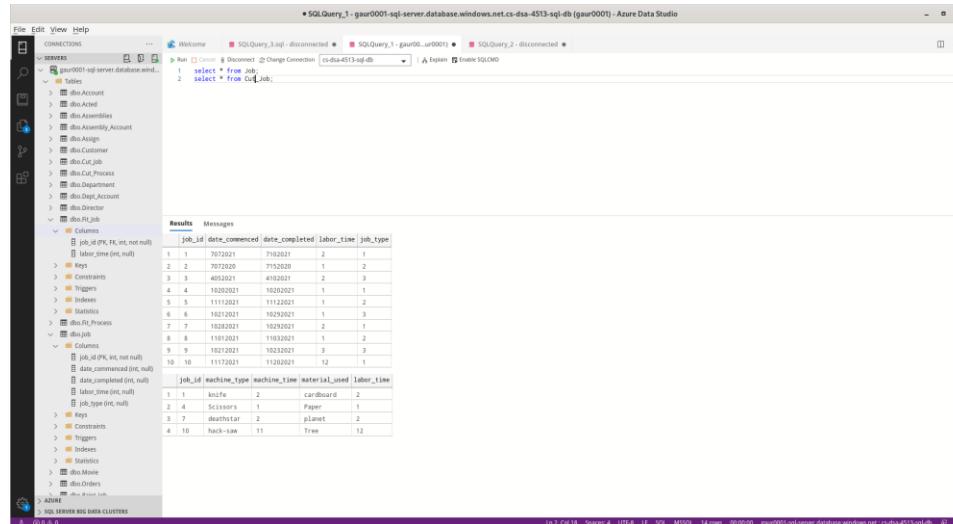


The screenshot shows the Eclipse IDE interface with the following details:

- File Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows packages AzureSql and HelloWorld.
- Code Editor:** Displays Java code for `HelloWorld.java`. The code includes a main method with logic to insert rows into a `Paint_Job` table and a loop to read from a `Customer` table. It also contains several comments numbered 6 through 20, likely corresponding to specific requirements or steps.
- Console Output:** Shows the output of running the application, including JDBC connection details and SQL queries being dispatched to the database. One query is shown as:


```
Dispatching the query ?(b)...  
Done. 1 row inserted in "Customer" table..  
Dispatching the query ?(b)...  
Done. 1 row inserted in "Paint_Job" table..
```
- Status Bar:** Shows the current state of the application.

Results:



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

- File Bar:** File, Edit, View, Help.
- Connections:** Shows a single connection to `gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001)`.
- Object Explorer:** Shows the database schema with tables like `Customer`, `Cut_Job`, `Paint_Job`, etc.
- Results Grid:** Displays the results of two queries:

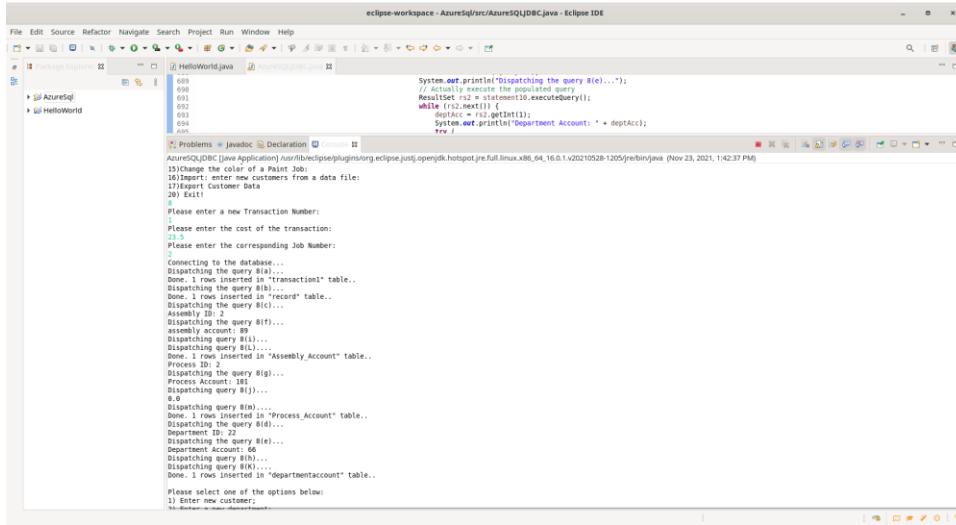

```
Results | Messages  
1 select * from Job;  
2 select * from Cut_Job;
```

job_id	Message	date_committed	date_completed	labor_time	job_type
1		7072021	7102021	2	1
2		7072020	7152020	1	2
3		4032021	4102021	2	3
4		1022021	1032021	1	1
5		11112021	11122021	1	2
6		10212021	10222021	1	3
7		1032021	10292021	2	1
8		11012021	11032021	1	2
9		10212021	10232021	3	3
10		11172021	11202021	12	1

job_id	Machine_Type	Machine_Time	Material_Used	Labor_Time
1	Knife	2	Cardboard	2
2	Scissors	1	Paper	1
3	Deathstar	2	Plastic	2
4	Hack-Saw	11	Tree	12
- Status Bar:** Shows the number of rows (2), columns (4), and execution time (14 ms).

Ten Queries of type 8:

First:



The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code editor displays Java code for the AzureSQL JDBC driver. The code includes imports for java.sql and org.json, and a main method that prints a message, creates a connection, and performs several database operations like inserting into 'transaction1' and 'record' tables, and updating 'Process_Account'. It also handles user input for transaction numbers and costs.

```

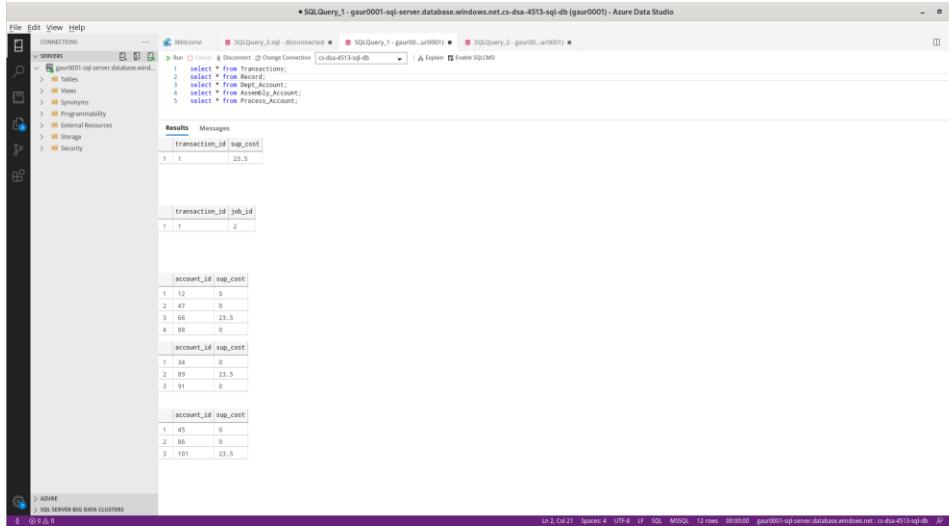
System.out.println("Dispatching the query B(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next()){
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
}
rs2.close();
}

Please enter a new Transaction Number:
Please enter the cost of the transaction:
23.5
Please enter the corresponding Job Number:
2
Connecting to the database...
Dispatching the query B(d)...
Done. 1 rows inserted in "transaction1" table.
Dispatching the query B(e)...
Done. 1 rows inserted in "record" table.
Dispatching the query B(f)...
Assembly ID: 1
Dispatching the query B(f)...
Assembly ID: 1
Dispatching query B(1)...
Dispatching query B(2)...
Dispatching query B(3)...
Dispatching query B(4)...
Dispatching query B(5)...
Process ID: 2
Dispatching the query B(g)...
Process Account: 181
Dispatching query B(1)...
8.9
Dispatching query B(n)...
Done. 1 rows inserted in "Process_Account" table..
Dispatching the query B(d)...
Department ID: 22
Dispatching query B(e)...
Department Account: 66
Dispatching query B(n)...
Done. 1 rows inserted in "departmentaccount" table..

Please select one of the options below:
1) Enter new customer;
2) Insert new record;
3) Insert new transaction;
4) Insert new job;
5) Insert new assembly;
6) Insert new department account;
7) Insert new process account;
8) Insert new record;
9) Insert new transaction;
10) Insert new job;
11) Insert new assembly;
12) Insert new department account;
13) Insert new process account;
14) Insert new record;
15) Change the color of a Paint Job;
16) Insert new customers from a data file;
17) Export customer data;
20) Exit!

```

Results:



The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The left sidebar shows connections to "gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db". The main area displays the results of four SQL queries:

- Query 1: "select * from Transactions;" - Result: transaction_id 1, sup_cost 23.5
- Query 2: "select * from Dept_Account;" - Result: account_id 1, job_id 1, sup_cost 2
- Query 3: "select * from Assembly_Account;" - Result: account_id 12, 47, 66, 88, sup_cost 0, 0, 23.5, 0
- Query 4: "select * from Process_Account;" - Result: account_id 34, 89, 91, sup_cost 0, 23.5, 0
- Query 5: "select * from departmentaccount;" - Result: account_id 45, 86, 101, sup_cost 0, 0, 23.5

Second:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace
  +-- AzureSql
    +-- HelloWorld
      +-- HelloWorld.java
      +-- resources
        +-- resources
          +-- resources
            +-- resources
              +-- resources
                +-- resources
                  +-- resources
                    +-- resources
                      +-- resources
                        +-- resources
                          +-- resources
                            +-- resources
                              +-- resources
                                +-- resources
                                  +-- resources
                                    +-- resources
                                      +-- resources
                                        +-- resources
                                          +-- resources
                                            +-- resources
                                              +-- resources
                                                +-- resources
                                                  +-- resources
                                                    +-- resources
                                                      +-- resources
                                                        +-- resources
                                                          +-- resources
                                                            +-- resources
                                                              +-- resources
                                                                +-- resources
                                                                  +-- resources
                                                                    +-- resources
                                                                      +-- resources
                                                                        +-- resources
                                                                          +-- resources
                                                                            +-- resources
                                                                              +-- resources
                                                                                +-- resources
                                                                                  +-- resources
                                                                                    +-- resources
                                                                                      +-- resources
                                                                                        +-- resources
                                                                                          +-- resources
                                                                                            +-- resources
                                                                                              +-- resources
                                                                                                +-- resources
                                                                                                  +-- resources
                                                                                                    +-- resources
                                                                                                      +-- resources
                                                                                                        +-- resources
                                                                                                          +-- resources
                                                                                                            +-- resources
................................................................
System.out.println("Dispatching the query #1(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next()){
  deptAcc = rs2.getInt(1);
  System.out.println("Department Account: " + deptAcc);
}
rs2.close();
System.out.println("Done. I rows inserted in " + "Assembly_Account" + " table..");
System.out.println("Done. I rows inserted in " + "Process_Account" + " table..");
System.out.println("Done. I rows inserted in " + "Dept_Account" + " table..");
System.out.println("Done. I rows inserted in " + "DepartmentAccount" + " table..");
System.out.println("Please select one of the options below:");
11 Enter new customer.

```

Results:

transaction_id	sup_cost
1	23.5
2	34.5

transaction_id	job_id
1	1
2	2
2	3

account_id	sup_cost
1	12
2	47
3	66
4	88

account_id	sup_cost
1	34
2	89
3	91

account_id	sup_cost
1	45
2	86
3	101

Third:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is a Java application using the AzureSQL JDBC driver. It includes imports for java.sql.* and org.apache.http.*. The code handles various database operations like inserting into 'transaction' and 'record' tables, and querying 'Dept_Account' and 'Dept' tables. A main loop asks for user input for transaction numbers and department IDs.

```

File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
src
AzureSQLDBC.java
AzureSQLDBC.java
System.out.println("Dispatching the query B(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next()){
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
    rs2.close();
}
}

// Please enter a new Transaction Number:
// Please enter the cost of the transaction:
// Please enter the corresponding Job Number:
// Connecting to the database...
// Dispatching the query B(a)...
Done. 1 rows inserted in "transaction" table.
// Dispatching the query B(c)...
Done. 1 rows inserted in "record" table.
Assembly ID: 1
Assembly ID: 1
// Dispatching the query B(f)...
Process completed.
// Dispatching the query B(g)...
// Dispatching the query B(h)...
Department ID: 44
// Dispatching the query B(i)...
// Dispatching query B(n)...
// Dispatching query B(m)...
Done. 1 rows inserted in "departmentaccount" table.
// Please select one of the options below:
11 Enter new customer.

```

Results:

The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The left sidebar shows connections to "gaur0001-sql-server.database.windows.net" and "gaur0001". The main area displays five tables with their data:

- Transactions:**

	transaction_id	sup_cost
1	1	23.5
2	2	34.5
3	3	77
- Job:**

	transaction_id	job_id
1	1	2
2	2	3
3	3	4
- Dept_Account:**

	account_id	sup_cost
1	12	0
2	47	34.5
3	66	23.5
4	88	77
- Dept:**

	account_id	sup_cost
1	34	0
2	89	23.5
3	91	34.5
- Customer:**

	account_id	sup_cost
1	45	0
2	86	34.5
3	101	23.5

Fourth:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
# | HelloWord.java | 1 HelloWord.java |
# | |
# | +-- AzureSql
# | +-- HelloWorld
# |
# | package AzureSql;
# |
# | import java.sql.*;
# |
# | public class HelloWord {
# |
# |     public static void main(String[] args) throws Exception {
# |
# |         System.out.println("Dispatching the query 8(e)...");
# |         // Actually execute the populated query
# |         ResultSet rs2 = statement10.executeQuery();
# |         while(rs2.next()){
# |             deptAcc = rs2.getInt(1);
# |             System.out.println("Department Account: "+ deptAcc);
# |             rs2.close();
# |         }
# |     }
# | }
# |
# | Problems [ javac ] Declaration [ Compiler ]
# | AzureSQLDBC [ java Application ] /src/org/eclipse/plugins/org.eclipse.jdt.openjdt.hotspot/pre_full/linux.x86_64_16.0.1.v20210528-1205/berlin/java (Nov 23, 2021, 1:42:37 PM)
# | 1) Enter new customer;
# | 2) Enter new department;
# | 3) Enter a new Process ;
# | 4) Enter a new Assembly;
# | 5) Enter a new Transaction Number;
# | 6) Enter a new Job;
# | 7) Enter the completion date of a Job;
# | 8) Enter a new Transaction Number;
# | 9) Retrieve the total cost incurred on an assembly-id
# | 10) Retrieve all accounts within a Department for a given day;
# | 11)Retriece the process through which a given assembly-id has passed;
# | 12)Retriece the department through which a job with a given number;
# | 13)Retriece the customers (in name order) whose category is in a given range;
# | 14)Delete all cut jobs with a Job Number;
# | 15)Display the current flight info;
# | 16)Import: enter new customers from a data file;
# | 17)Import: Customer Data
# | 18) Exit!
# |
# | Please enter a new Transaction Number:
# | 4
# | Please enter the cost of the transaction:
# | 10
# | Please enter the corresponding Job Number:
# | 1
# |
# | Connecting to the database...
# | Dispatching the query 8(f)...
# | Done. 1 rows inserted in "transaction" table.
# | Dispatching the query 8(b)...
# | Done. 1 rows inserted in "dept" table.
# | Dispatching the query 8(c)...
# | Done. 1 rows inserted in "deptacct" table.
# | Dispatching the query 8(f)...
# | Process ID: 5
# | Dispatching the query 8(g)...
# | Dispatching the query 8(d)...
# | Dispatching the query 8(e)...
# |
# | Please select one of the options below:
# | 1) Enter new customer...
```

Results:

	transaction_id	sup_cost
1	1	23.5
2	2	34.5
3	3	77
4	4	76

	transaction_id	job_id
1	1	2
2	2	3
3	3	4
4	4	5

	account_id	sup_cost
1	12	0
2	47	34.5
3	66	23.5
4	88	77

	account_id	sup_cost
1	34	0
2	89	23.5
3	91	34.5

	account_id	sup_cost
1	45	0
2	86	34.5
3	101	23.5

Fifth:

Results:

The screenshot shows the Azure Data Studio interface with a query results grid. The grid has three rows of data, each representing a different table:

	transaction_id	sup_cost
1	1	23.5
2	2	34.5
3	3	77
4	4	76
5	10	66.5

	transaction_id	job_id
1	1	2
2	2	3
3	3	4
4	4	5
5	10	8

	account_id	sup_cost
1	12	0
2	47	101
3	66	23.5
4	88	77

Below the grid, there are three small tables with the same structure:

	account_id	sup_cost
1	34	0
2	89	23.5
3	91	34.5

	account_id	sup_cost
1	45	0
2	86	34.5
3	101	23.5

	account_id	sup_cost
1	45	0
2	86	34.5
3	101	23.5

Sixth:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace
  +-- AzureSQLDBC [java Application]
    +-- AzureSQLDBC.java
    +-- HelloWorld
      +-- HelloWorld.java
      +-- HelloWorld.pom
      +-- HelloWorld.war

Problems | javadoc | Declaration | Content

System.out.println("Dispatching the query #1(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next()){
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
}
rs2.close();

Please enter a new Transaction Number:
1
Please enter the cost of the transaction:
10
Please enter the corresponding Job Number:
1
Connecting to the database...
Dispatching the query #1(a)...
Done. 1 rows inserted in "transaction" table..
Dispatching the query #1(b)...
Done. 1 rows inserted in "process_record" table..
Dispatching the query #1(c)...
Assembly ID: 3
Dispatching the query #1(f)...
assembly account: #1
Dispatching query #1(g)...
Dispatching query #1(h)...
Done. 1 rows inserted in "Assembly_Account" table..
Process Account inserted...
Dispatching the query #1(g)...
Dispatching query #1(h)...
34.
Dispatching query #1(i)...
Done. 1 rows inserted in "Process_Account" table..
Dispatching query #1(j)...
Assembly ID: 33
Dispatching the query #1(k)...
Done. 1 rows inserted in "DepartmentAccount" table..
Done. 1 rows inserted in "departmentaccount" table..

Please select one of the options below:
1. Insert a new record
2. Update an existing record
3. Delete an existing record
4. Exit

```

Results:

transaction_id	job_id	cost
1	1	20.5
2	2	34.5
3	3	77
4	4	76
5	10	66.5
6	11	53.1

transaction_id	job_id
1	1
2	2
3	3
4	4
5	10
6	11

account_id	sup_cost
1	12
2	47
3	68
4	88
5	10
6	11

account_id	sup_cost
1	34
2	89
3	91

Seventh:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
089
090
091
092
093
094
095
System.out.print("Dispatching the query B(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next())
{
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
    rs2.close();
}
System.out.println("Done. 1 row inserted in \"transaction\" table..");
System.out.println("Done. 1 rows inserted in \"record\" table..");
Assembly ID: 44
Disposing query B(f)...
PreparedStatement ps;
Disposing query B(g)...
Disposing query B(h)...
Disposing query B(i)...
Disposing query B(j)...
Disposing query B(k)...
Done. 1 row inserted in "departmentaccount" table..
11. Enter new customer...
Please select one of the options below:
11. Enter new customer...

```

Results:

dept_id	dept_name
3	Customer Service
4	Public Relations
5	Marketing
6	Sales
7	Administrative

transaction_id	job_id	sup_cost
1	1	2
2	2	3
3	3	4
4	4	5
5	10	8
6	11	3
7	13	43.1

account_id	sup_cost
12	0
47	154.5
66	23.5
88	100.1

account_id	sup_cost
34	0
89	23.5
91	87.6

Eight:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | HelloWord.java | Resources.java |
089
090
091
092
093
094
095
System.out.print("Dispatching the query #1(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next())
{
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
    rs2.close();
}
System.out.println("Done. 1 rows inserted in \"transaction\" table..");
Disposing the query #1(f)...
Process ID: 5
Disposing the query #1(g)...
Disposing the query #1(h)...
Departments: 55
Disposing the query #1(i)...
Please select one of the options below:
11. Enter new customer;
12. Enter a new Process ;
13. Enter a new Assembly;
14. Enter a new Job;
15. Enter a new Transaction Number;
16. Import: enter new customers from a data file;
17. Import Customer Data
18. Exit
Please enter a new Transaction Number:
14
Please enter the cost of the transaction:
15
Please enter the corresponding Job Number:
1
Connecting to the database...
Connecting to the database...
Done. 1 rows inserted in "transaction" table..
Disposing the query #1(b)...
Done. 1 rows inserted in "dept" table..
Disposing the query #1(c)...
Disposing the query #1(d)...
Disposing the query #1(e)...
Process ID: 5
Disposing the query #1(g)...
Disposing the query #1(h)...
Departments: 55
Disposing the query #1(i)...
Please select one of the options below:
11. Enter new customer;
12. Enter a new Process ;
13. Enter a new Assembly;
14. Enter a new Job;
15. Enter a new Transaction Number;
16. Import: enter new customers from a data file;
17. Import Customer Data
18. Exit

```

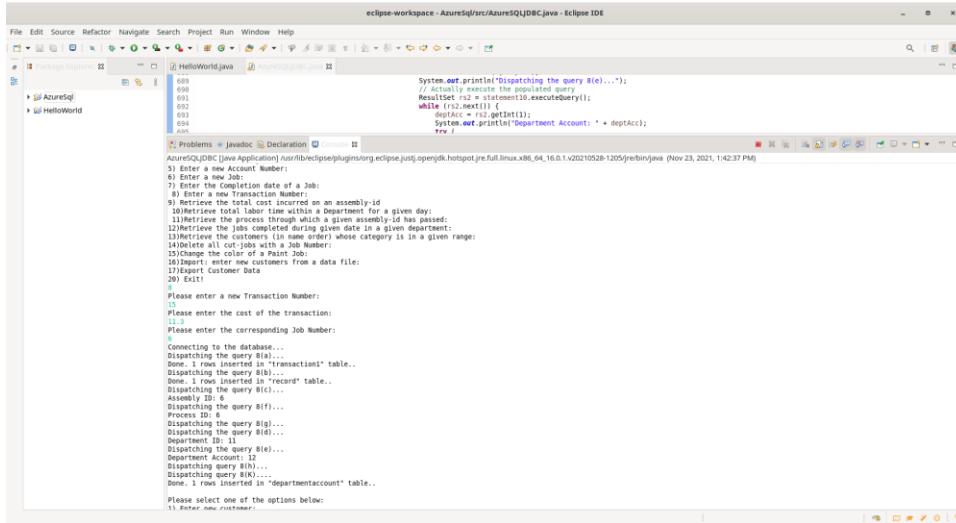
Results:

transaction_id	job_id
5	10
5	11
2	13
8	14

account_id	sup_cost
1	12
2	47
3	66
4	88

account_id	sup_cost
1	34
2	89
3	91

Ninth:

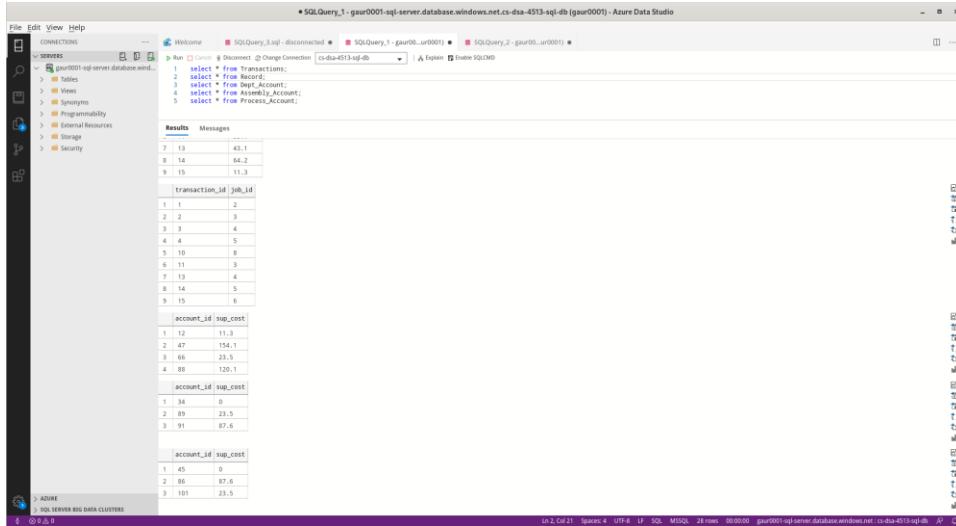


The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code editor displays Java code for a JDBC application:

```

    689     System.out.print("Dispatching the query B(e)...");
    690     // Actually execute the populated query
    691     ResultSet rs2 = statement10.executeQuery();
    692     while(rs2.next()){
    693         deptAcc = rs2.getInt(1);
    694         System.out.println("Department Account: "+ deptAcc);
    695     }
    696 }
    697
    698 // Please enter a new Transaction Number:
    699 // 1) Enter a new Transaction Number:
    700 // 2) Enter the completion date of a Job:
    701 // 3) Enter the completion date of a Job:
    702 // 4) Enter a new Transaction Number:
    703 // 5) Enter a new Customer Number:
    704 // 6) Enter a new Customer Number:
    705 // 7) Enter the completion date of a Job:
    706 // 8) Enter a new Transaction Number:
    707 // 9) Enter a new Customer Number:
    708 // 10) Retrieve total labor time within a Department for a given day:
    709 // 11) Retrieve the number of days a job has been completed since its start date:
    710 // 12) Retrieve the jobs completed during given date in a given department:
    711 // 13) Retrieve the customers (in name order) whose category is in a given range:
    712 // 14) Retrieve the number of hours worked by a customer:
    713 // 15) Change the color of a Paint job:
    714 // 16) Enter new customers from a data file:
    715 // 17) Export customer data:
    716 // 20) Exit
    717
    718 // Please enter a new Transaction Number:
    719 // 1)
    720 // Please enter the cost of the transaction:
    721 // 11)
    722 // Please enter the corresponding Job Number:
    723 Connecting to the database...
    724 Dispatching the query B(f)...
    725 Done. 1 rows inserted in "transaction" table.
    726 Dispatching the query B(g)...
    727 Done. 1 rows inserted in "record" table.
    728 Assembly ID: 3
    729 Dispatching the query B(h)...
    730 Done. 1 rows inserted in "record" table.
    731 Dispatching the query B(i)...
    732 Done. 1 rows inserted in "record" table.
    733 Dispatching the query B(j)...
    734 Done. 1 rows inserted in "record" table.
    735 Dispatching the query B(k)...
    736 Done. 1 rows inserted in "record" table.
    737 Dispatching the query B(l)...
    738 Done. 1 rows inserted in "record" table.
    739 Dispatching the query B(m)...
    740 Done. 1 rows inserted in "record" table.
    741 Dispatching the query B(n)...
    742 Done. 1 rows inserted in "record" table.
    743 Dispatching the query B(o)...
    744 Done. 1 rows inserted in "record" table.
    745 Dispatching the query B(p)...
    746 Done. 1 rows inserted in "record" table.
    747 Dispatching the query B(q)...
    748 Done. 1 rows inserted in "record" table.
    749 Dispatching the query B(r)...
    750 Done. 1 rows inserted in "record" table.
    751 Dispatching the query B(s)...
    752 Done. 1 rows inserted in "record" table.
    753 Dispatching the query B(t)...
    754 Done. 1 rows inserted in "record" table.
    755 Dispatching the query B(u)...
    756 Done. 1 rows inserted in "record" table.
    757 Dispatching the query B(v)...
    758 Done. 1 rows inserted in "record" table.
    759 Dispatching the query B(w)...
    760 Done. 1 rows inserted in "record" table.
    761 Dispatching the query B(x)...
    762 Done. 1 rows inserted in "record" table.
    763 Dispatching the query B(y)...
    764 Done. 1 rows inserted in "record" table.
    765 Dispatching the query B(z)...
    766 Done. 1 rows inserted in "record" table.
    767
    768 Please select one of the options below:
    769 1) Enter new customer:
    
```

Results:



The screenshot shows the Azure Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The results pane displays the output of several SQL queries:

- transaction** table:

transaction_id	job_id
7	12
8	14
9	15
10	13
11	1
12	2
13	3
14	4
15	5
16	6
17	7
18	8
19	9
20	10
21	11
22	12
23	13
24	14
25	15
- account** table:

account_id	sup_cost
1	12
2	47
3	64
4	23.5
5	100.1
- customer** table:

customer_id	sup_cost
1	34
2	89
3	91

Tenth:

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code in the editor is for a Java application using the Azure JDBC driver. The output window displays the execution of the code, showing the process of inserting data into tables like 'transaction', 'dept', 'customer', and 'deptaccount'. It includes prompts for transaction number, cost, and job number, and lists options for new customer, assembly, account, or assembly number.

```

System.out.print("Dispatching the query 8(e)...");
// Actually execute the populated query
ResultSet rs2 = statement10.executeQuery();
while(rs2.next()){
    deptAcc = rs2.getInt(1);
    System.out.println("Department Account: " + deptAcc);
}
rs2.close();
System.out.println("Done. " + rowsInserted + " rows inserted in " + "transaction" + " table..");
System.out.println("Done. " + rowsInserted + " rows inserted in " + "dept" + " table..");
System.out.println("Done. " + rowsInserted + " rows inserted in " + "customer" + " table..");
System.out.println("Done. " + rowsInserted + " rows inserted in " + "deptaccount" + " table..");
Please enter a new Transaction Number:
Please enter the cost of the transaction:
Please enter the corresponding Job Number:
1. Enter a new customer;
2. Enter a new department;
3. Enter a new assembly;
4. Enter a new Assembly;
5. Enter a new Account Number;

```

Results:

The screenshot shows the Microsoft Data Studio interface with the title "SQLQuery_1 - gaur0001-sql-server.database.windows.net.cs-dsa-4513-sql-db (gaur0001) - Azure Data Studio". The results pane displays the output of three SQL queries. The first query shows data from the 'transaction' table. The second query shows data from the 'dept' table. The third query shows data from the 'deptaccount' table. The results are presented as tables with columns such as transaction_id, job_id, account_id, and sup_cost.

transaction_id	job_id
7	13
8	14
9	15
10	17

dept_id	dept_name
1	Research
2	Development
3	Marketing
4	Sales
5	Customer Service
6	Public Relations
7	Human Resources
8	Manufacturing
9	Quality Control

dept_id	dept_name	account_id	sup_cost
1	Research	34	0
2	Development	47	154.1
3	Marketing	66	23.5
4	Sales	88	189
5	Customer Service	91	67.0
6	Public Relations	92	73.0
7	Human Resources	93	11.3
8	Manufacturing	94	23.5
9	Quality Control	101	23.5

Three Queries of Type 10

First:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the code for `HelloWorld.java`. The code includes logic to calculate total labor time for completed jobs across departments.
- Terminal Output:** Shows the execution of the Java application. It prints a menu of 17 options related to database operations like inserting new accounts, retrieving departmental data, and deleting specific records.
- Bottom Terminal Line:** The user has selected option 11, which asks for a department number and a completion date. The terminal shows the user input and the system's response: "The total labor time within the given 11 department for jobs completed during the 11112021 date is: 0 hours."

Second:

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the code for `HelloWorld.java`.
- Terminal Output:** Shows the execution of the Java application. It prints a menu of 17 options.
- Bottom Terminal Line:** The user has selected option 11, which asks for a department number and a completion date. The terminal shows the user input and the system's response: "The total labor time within the given 11 department for jobs completed during the 11112021 date is: 0 hours."

Third:

Screenshot of Eclipse IDE showing the code editor for `HelloWorld.java` and a terminal window. The code performs a database query to calculate total labor time for completed jobs across departments. The terminal window shows the results of the query and a list of numbered options for the user to select from.

```

HelloWorld.java
...
168     while(rs2.next()) {
169         larray[i] = rs2.getInt("labor_time");
170         darray[i] = rs2.getInt("date_completed");
171     }
172     int totalLaborTime = 0;
173     for (int i = 0; i < lindex; i++) {
174         if(darray[i] == dateCompleted) {
175             totalLaborTime += larray[i];
176         }
177     }
178     System.out.println("The total labor time within the given " + deptNo + " department for jobs completed during the " + dateCompleted + " date is: " + totalLaborTime + " hours.");
179     //System.out.println(String.format("Done. %d rows inserted in 'customers' table.", rowsInserted));
180 }
181
182
183     Please enter the department number for determining the labor cost:
184     Please enter the date of completion for the jobs:
185 10212621
186 Dispatching the query 1...
187     Dispatching the query 1(a)...
188     Dispatching the query 1(b)...
189     Dispatching the query 1(c)...
190     Dispatching the query 1(c)...
191     Dispatching the query 1(c)...
192     The total labor time within the given 33 department for jobs completed during the 10212621 date is: 0 hours.
193
194     Please select one of the options below:
195     1) Enter new customer;
196     2) Enter a new department;
197     3) Enter a new Process;
198     4) Enter a new Assembly;
199     5) Enter a new Account Number;
200     6) Enter a new Job;
201

```

Three Queries of Type 11

First:

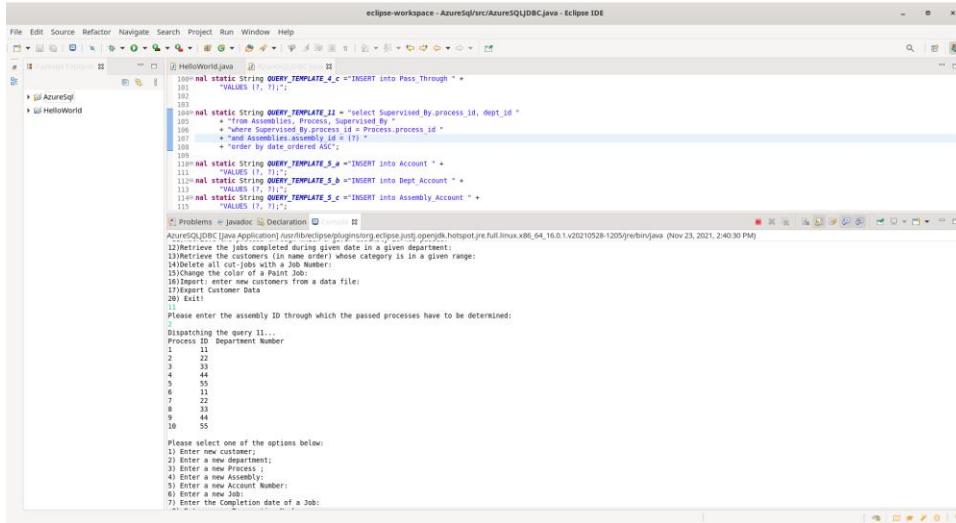
Screenshot of Eclipse IDE showing the code editor for `HelloWorld.java` and a terminal window. The code contains three static string variables representing SQL INSERT queries into Pass_Through, Account, and Assembly_Account tables. The terminal window shows a numbered list of options for the user to select from.

```

HelloWorld.java
...
160     nal static String QUERY_TEMPLATE_4_c ="INSERT into Pass_Through "
161     "VALUES (17, ?);"
162
163     nal static String QUERY_TEMPLATE_5_a = "select Supervised_By.process_id,dept_id "
164     "from Pass_AssemblyLine_Proc_Supervised_By "
165     "+ where Supervised_By.process_id = ? "
166     "+ and Pass_AssemblyLine_Proc_Supervised_By.id = ? "
167     "+ and Pass_AssemblyLine_Proc_Supervised_By.dept_id = ? "
168
169     nal static String QUERY_TEMPLATE_5_b = "INSERT into Dept_Account "
170     "VALUES (17, ?);"
171     nal static String QUERY_TEMPLATE_5_c = "INSERT into Assembly_Account "
172     "VALUES (17, ?);"
173
174     Please select one of the options below:
175     1) Enter new Transaction Number;
176     2) Retrieve the total cost incurred on an assembly-line by a process id;
177     3) Retrieve the process through which a given assembly-line has passed;
178     4) Retrieve the jobs completed during given date in a given department;
179     5) Delete all cut-jobs with a Job Number;
180     6) Change the color of a Paint Job;
181     7) Enter new customers From a data file;
182     8) Expert Customer Data;
183
184     Please enter the assembly ID through which the passed processes have to be determined:
185     Dispatching The query 11...
186     Process ID Department Number
187     1 11
188     2 22
189     3 33
190     4 44
191     5 55
192     6 11
193     7 22
194     8 33
195     9 44
196     10 55
197
198     Please select one of the options below:
199     1) Enter new customer;
200     2) Enter a new department;
201     3) Enter a New Process;
202     4) Passes An Assembly Line

```

Second:

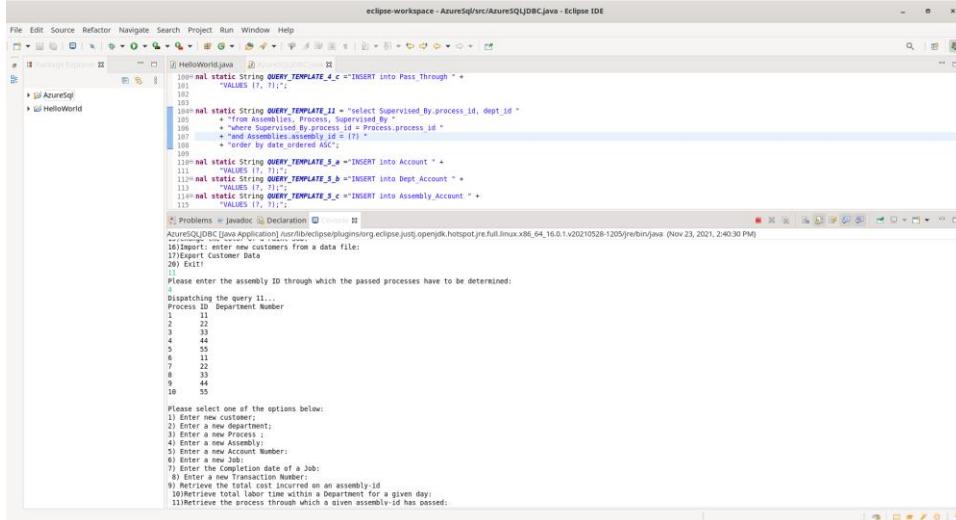


The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE". The code editor displays Java code for an AzureSQL JDBC application. The code includes several static String variables representing SQL queries for inserting data into tables like Pass_Through, Account, Dept_Account, and Assembly_Account. A series of numbered comments (1 through 15) explain the purpose of each query. Below the code, a message box is displayed with the following text:

```

    Problems = javadoc Declaration
    Eclipse IDE for Java Application [usr/lib/eclipse/plugins/org.eclipse.just/jopenjdk.hotspot.jre.full/linux.x86_64_16.0.1.v20210528-1205/rebin/java] (Nov 23, 2021, 2:40:30 PM)
    12) Retrieve the total cost incurred during given date in a given department;
    13) Retrieve the customers (in name order) whose category is in a given range;
    14) Delete all cut jobs with a Job Number;
    15) Enter a new Customer Data;
    16) Import: enter new customers from a data file;
    17) Enter Customer Data;
    20) Exit;
    ...
    Please enter the assembly ID through which the passed processes have to be determined:
    Dispatching the query 11...
    Process ID Department Number
    1 11
    2 22
    3 33
    4 44
    5 55
    6 11
    7 22
    8 33
    9 44
    10 55
    Please select one of the options below:
    1) Enter new customer;
    2) Enter a new customer;
    3) Enter a new Process;
    4) Enter a new Assembly;
    5) Enter a new Transaction Number;
    6) Enter a new Job;
    7) Enter the Completion date of a Job;
    ...
  
```

Third:



The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE". The code editor displays Java code for an AzureSQL JDBC application. The code includes several static String variables representing SQL queries for inserting data into tables like Pass_Through, Account, Dept_Account, and Assembly_Account. A series of numbered comments (1 through 15) explain the purpose of each query. Below the code, a message box is displayed with the following text:

```

    Problems = javadoc Declaration
    Eclipse IDE for Java Application [usr/lib/eclipse/plugins/org.eclipse.just/jopenjdk.hotspot.jre.full/linux.x86_64_16.0.1.v20210528-1205/rebin/java] (Nov 23, 2021, 2:40:30 PM)
    12) Retrieve the total cost incurred during given date in a given department;
    13) Retrieve the customers (in name order) whose category is in a given range;
    14) Delete all cut jobs with a Job Number;
    15) Enter a new Customer Data;
    16) Import: enter new customers from a data file;
    17) Enter Customer Data;
    20) Exit;
    ...
    Please enter the assembly ID through which the passed processes have to be determined:
    Dispatching the query 11...
    Process ID Department Number
    1 11
    2 22
    3 33
    4 44
    5 55
    6 11
    7 22
    8 33
    9 44
    10 55
    Please select one of the options below:
    1) Enter new customer;
    2) Enter a new customer;
    3) Enter a new Process;
    4) Enter a new Assembly;
    5) Enter a new Transaction Number;
    6) Enter a new Job;
    7) Enter the Completion date of a Job;
    8) Enter a new Transaction Number;
    9) Retrieve the total cost incurred on an assembly-id;
    10) Retrieve total labor cost within a department for a given day;
    11) Retrieves the process threads which a given assembly-id has passed;
  
```

Three Queries of Type 12

First

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
- Left Sidebar:** Shows the project structure with files like `HelloWorld.java`, `AzureSql`, and `HelloWorld`.
- Code Editor:** Displays Java code for `HelloWorld.java` containing static string templates for various database queries.
- Console:** Shows the output of a command-line application running on Linux, displaying a menu of 20 options related to database operations like inserting into tables, selecting data, and deleting records.
- Status Bar:** Shows the path `AzureSQLJDBC.java Application` and the date/time `Nov 23, 2021, 2:40:30 PM`.

Second:

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
- Left Sidebar:** Shows the project structure with files like `HelloWorld.java`, `AzureSql`, and `HelloWorld`.
- Code Editor:** Displays Java code for `HelloWorld.java` containing static string templates for various database queries.
- Console:** Shows the output of a command-line application running on Linux, displaying a menu of 20 options related to database operations like inserting into tables, selecting data, and deleting records.
- Status Bar:** Shows the path `AzureSQLJDBC.java Application` and the date/time `Nov 23, 2021, 2:40:30 PM`.

Third:

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code editor displays Java code for three static string variables:

```

108    final static String QUERY_TEMPLATE_4_c = "INSERT into Pass_Through " +
109        "VALUES (?)";
110
111    final static String QUERY_TEMPLATE_12_a = "select Supervised_By.process_id, dept_id " +
112        "from Assemblies,Process_Supervised_By " +
113        "where Supervised_By.process_id = Process.process_id " +
114        "and Assemblies.assembly_id = (?) ";
115
116    final static String QUERY_TEMPLATE_5_a = "INSERT into Account " +
117        "VALUES (?)";
118
119    final static String QUERY_TEMPLATE_5_b = "INSERT into Dept_Account " +
120        "VALUES (?)";
121
122    final static String QUERY_TEMPLATE_5_c = "INSERT into Assembly_Account " +
123        "VALUES (?)";

```

The status bar at the bottom indicates: "AzureSQLDBC [Java Application] /src/AzureSQLDBC.java - Eclipse IDE" and "Nov 23, 2021, 2:40:30 PM". Below the code editor, a message box lists 20 numbered options for interacting with the database.

Three Queries of Type 13

First

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE". The code editor displays Java code for three static string variables:

```

108    final static String QUERY_TEMPLATE_4_c = "INSERT into Pass_Through " +
109        "VALUES (?)";
110
111    final static String QUERY_TEMPLATE_12_a = "select Supervised_By.process_id, dept_id " +
112        "from Assemblies,Process_Supervised_By " +
113        "where Supervised_By.process_id = Process.process_id " +
114        "and Assemblies.assembly_id = (?) ";
115
116    final static String QUERY_TEMPLATE_5_a = "INSERT into Account " +
117        "VALUES (?)";
118
119    final static String QUERY_TEMPLATE_5_b = "INSERT into Dept_Account " +
120        "VALUES (?)";
121
122    final static String QUERY_TEMPLATE_5_c = "INSERT into Assembly_Account " +
123        "VALUES (?)";

```

The status bar at the bottom indicates: "AzureSQLDBC [Java Application] /src/AzureSQLDBC.java - Eclipse IDE" and "Nov 23, 2021, 2:40:30 PM". Below the code editor, a message box lists 20 numbered options for interacting with the database.

Second

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE". The code editor displays Java code for the AzureSQLJDBC class, specifically static String variables QUERY_TEMPLATE_4_c, QUERY_TEMPLATE_12, QUERY_TEMPLATE_5_a, QUERY_TEMPLATE_5_b, and QUERY_TEMPLATE_5_c, which contain SQL INSERT statements. Below the code editor is a terminal window showing the output of a database query:

```

AzureSQLJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justmyjava/jdk/hotspot/re.full/linux.x86_64_16.0.1/v20210528-1205/rebin/java (Nov 23, 2021, 2:40:30 PM)
4) Enter a new Assembly:
5) Enter a new Account Number:
6) Enter a new Department:
7) Enter the Completion date of a Job:
8) Enter a new Transaction Number:
9) Retrieve the total labor time on an assembly-id
10)Retrieve total labor time within a Department for a given Day:
11)Enter the process group for a job that has passed:
12)Retrieve the completed during a date in a given department:
13)Retrieve the customers (in same order) whose category is in a given range:
14)Change the color of a Paint Job:
15)Change the color of a Paint Job:
16)Import: enter new customers from a data file:
17)Import Customer Data
20) Exit!
Please input the minimum category:
Please input the maximum category:
Connecting to the database...
Dispatching the query...
Dispatching the query 13:
Customer [CustomerID | CustomerName | Category]:
Name | Address | Category
-----|-----|-----
John | Norma | 2
Richard | Moore | 5
Dwight | Schrute | 4
Steve | Edmond | 3
Please select one of the actions below:

```

Third

The screenshot shows the Eclipse IDE interface with the title "eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE". The code editor displays Java code for the AzureSQLJDBC class, specifically static String variables QUERY_TEMPLATE_4_c, QUERY_TEMPLATE_12, QUERY_TEMPLATE_5_a, QUERY_TEMPLATE_5_b, and QUERY_TEMPLATE_5_c, which contain SQL INSERT statements. Below the code editor is a terminal window showing the output of a database query:

```

AzureSQLJDBC [Java Application] /usr/lib/eclipse/plugins/org.eclipse.justmyjava/jdk/hotspot/re.full/linux.x86_64_16.0.1/v20210528-1205/rebin/java (Nov 23, 2021, 2:40:30 PM)
1) Exit!
2) Enter a new department;
3) Enter a new Process ;
4) Enter a new customer;
5) Enter a new Account Number;
6) Enter a new Transaction Number;
7) Enter the completion date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total labor time on an assembly-id
10)Retrieve the total labor time within a Department for a given day:
11)Enter the process group for a job that has passed:
12)Retrieve the completed during a date in a given department:
13)Retrieve the customers (in same order) whose category is in a given range:
14)Change the color of a Paint Job:
15)Change the color of a Paint Job:
16)Import: enter new customers from a data file:
17)Import Customer Data
20) Exit!
Please input the minimum category:
Please input the maximum category:
Connecting to the database...
Dispatching the query...
Dispatching the query 13:
Customer [CustomerID | CustomerName | Category]:
Name | Address | Category
-----|-----|-----
John | Norma | 2
Richard | Moore | 5
Dwight | Schrute | 4
Steve | Edmond | 3
Please select one of the actions below:

```

One Query of Type 14

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | 1 | HelloWord.java | 2 |
+-- AzureSql
+-- HelloWorld
1 package AzureSql;
2 import java.sql.*;
3
4 public class HelloWord {
5     public static void main(String[] args) {
6         String url = "jdbc:sqlserver://127.0.0.1:1433;databaseName=master";
7         String user = "sa";
8         String password = "P@ssw0rd";
9
10        Connection conn = null;
11        Statement st = null;
12
13        try {
14            conn = DriverManager.getConnection(url, user, password);
15            st = conn.createStatement();
16
17            //1)Delete all cut-jobs with a Job Number:
18            String QUERY_TEMPLATE_4_a = "DELETE FROM [dbo].[cut] WHERE job_number = ? ";
19            String QUERY_TEMPLATE_4_b = "DELETE FROM [dbo].[job] WHERE job_number = ? ";
20            String QUERY_TEMPLATE_4_c = "DELETE FROM [dbo].[painsje] WHERE job_number = ? ";
21
22            PreparedStatement ps = st.prepareStatement(QUERY_TEMPLATE_4_a);
23            ps.setInt(1, 17);
24            ps.executeUpdate();
25
26            PreparedStatement ps2 = st.prepareStatement(QUERY_TEMPLATE_4_b);
27            ps2.setInt(1, 17);
28            ps2.executeUpdate();
29
30            PreparedStatement ps3 = st.prepareStatement(QUERY_TEMPLATE_4_c);
31            ps3.setInt(1, 17);
32            ps3.executeUpdate();
33
34            System.out.println("Done, 1 row updated in \"painsje\" table..");
35        } catch (SQLException e) {
36            e.printStackTrace();
37        }
38    }
39 }

```

Please enter the minimum Job Number:
17
Please enter the maximum Job Number:
17
Connecting to the database...
Dispatching the query 14(a)...
Done, 1 rows updated in "cut" table...
Dispatching the query 14(b)...
Done, 1 rows updated in "job" table...
Dispatching the query 14(c)...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The DELETE statement conflicted with the REFERENCE constraint "FK_Record_job_id_8829727". The conflict occurred in database "master". The statement has been terminated.
at com.microsoft.sqlserver.jdbc.JDBCUtil.throwException(JDBCUtil.java:262)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.getMoreResults(SQLServerPreparedStatement.java:1662)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:615)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:537)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:533)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:348)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:262)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:227)
at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:188)
at AzureSQLJDBC.main(AzureSQLJDBC.java:138)

This execution also shows the error handling being done by Azure SQL Server itself. In this case, a foreign key constraint is being violated.

One Query of Type 15

First:

```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Windows Help
# | Package Explorer | 1 | HelloWord.java | 2 |
+-- AzureSql
+-- HelloWorld
1 package AzureSql;
2 import java.sql.*;
3
4 public class HelloWord {
5     public static void main(String[] args) {
6         String url = "jdbc:sqlserver://127.0.0.1:1433;databaseName=master";
7         String user = "sa";
8         String password = "P@ssw0rd";
9
10        Connection conn = null;
11        Statement st = null;
12
13        try {
14            conn = DriverManager.getConnection(url, user, password);
15            st = conn.createStatement();
16
17            System.out.println("System.out.println(\"Done, " + rows_inserted + " rows deleted in " + table + ".\");");
18            System.out.println("final int jobno = sc.nextInt();");
19            System.out.println("final String color = sc.nextLine();");
20
21            st.executeUpdate("DELETE FROM [dbo].[cut] WHERE job_number = " + jobno);
22            st.executeUpdate("UPDATE [dbo].[job] SET color = " + color + " WHERE job_number = " + jobno);
23
24            st.executeUpdate("SELECT COUNT(*) AS total FROM [dbo].[cut] WHERE job_number = " + jobno);
25            st.executeUpdate("SELECT COUNT(*) AS total FROM [dbo].[job] WHERE job_number = " + jobno);
26
27            System.out.println("Done, 1 row updated in " + table + " table..");
28        } catch (SQLException e) {
29            e.printStackTrace();
30        }
31
32        System.out.println("Please enter the Job Number for the color change:");
33        final int jobno = sc.nextInt();
34
35        System.out.println("Please enter the new color for the job:");
36        final String color = sc.nextLine();
37
38        try {
39            st.executeUpdate("UPDATE [dbo].[job] SET color = " + color + " WHERE job_number = " + jobno);
40        } catch (SQLException e) {
41            e.printStackTrace();
42        }
43
44        System.out.println("Done, 1 row updated in " + table + " table..");
45    }
46 }

```

Please enter one of the options below:
1) Enter new customer;
2) Enter new job;
3) Enter a new Process;
4) Enter a new Assembly;
5) Enter a new Account Number;
6) Enter a new Job;
7) Enter the transaction date of a Job;
8) Enter a new Transaction Number;
9) Retrieve the total count of time within an assembly_id;
10) Retrieve the total count of time within a given day;
11) Retrieve the process through which a given assembly_id has passed;
12) Retrieve the assembly_id of a given process;
13) Retrieve the customer(s) (in name order) whose category is in a given range;
14) Delete all cut-jobs with a Job Number;
15) Change the color of a Paint Job;
16) Import: enter new customers from a data file;
17) Import Customer Data;
18) Exit;
19) Please enter the Job Number for the color change;
20) Please enter the new color for the job:
CarneMan
Connecting to the database...
Dispatching the query 15(a)...
Done, 1 row updated in "painsje" table..

Second:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | [HelloWorld.java]
1343     }
1344     System.out.println(String.format("Done. %d rows deleted in \"job\" table..", rows_inserted));
1345     break;
1346   case '1':
1347     System.out.println("Please enter the Job Number for the color change:");
1348     final int jobno = sc.nextInt();
1349     final String color = sc.nextLine();
1350     System.out.println("Please enter the new color for the job:");
1351     final String newcolor = sc.nextLine();
1352     System.out.println("Connecting to the database..."); // Get the database connection, create statement and execute it right away, as no user input need be collected
1353     // try {
1354     //   final Connection connection = DriverManager.getConnection(url);
1355     // } catch (SQLException e) {
1356     //   e.printStackTrace();
1357   } catch (InputMismatchException e) {
1358     System.out.println("Please select one of the options below:");
1359     // Enter new customer;
1360     // Enter a new department;
1361     // Enter a new Process;
1362     // Enter a new Assembly;
1363     // Enter a new Account Number;
1364     // Enter a new Job;
1365     // Enter the completion date of a Job;
1366     // Enter a new transaction Number;
1367     // Retrieve the total cost incurred on an assembly-id;
1368     // Delete all jobs completed on a Department for a given day;
1369     // Retrieve the total labor hours worked on a Department for a given day;
1370     // Retrieve the total labor hours worked on a Department for a given month;
1371     // Retrieve the total labor hours worked on a Department for a given year;
1372     // Retrieve the jobs completed during given date in a given department;
1373     // Retrieve the jobs completed during given month in a given department;
1374     // Retrieve the jobs completed during given year in a given department;
1375     // Delete all cut-jobs with a Job Number;
1376     // Change the color of a Paint Job;
1377     // Import: enter new customers from a data file;
1378     // Export Customer Data;
1379     // Exit!
1380     System.out.println("Please enter the Job Number for the color change:");
1381     final int jobno = sc.nextInt();
1382     final String color = sc.nextLine();
1383     System.out.println("Please enter the new color for the job:");
1384     final String newcolor = sc.nextLine();
1385     System.out.println("Connecting to the database..."); // Dispatching the query 15...
1386     // Dispatching the query 15...
1387     // Done. 1 rows updated in "paintjob" table..
1388   }

```

Third:

```

eclipse-workspace - AzureSql/src/AzureSQLDBC.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
# | Package Explorer | [HelloWorld.java]
1343     }
1344     System.out.println(String.format("Done. %d rows deleted in \"job\" table..", rows_inserted));
1345     break;
1346   case '1':
1347     System.out.println("Please enter the Job Number for the color change:");
1348     final int jobno = sc.nextInt();
1349     final String color = sc.nextLine();
1350     System.out.println("Please enter the new color for the job:");
1351     final String newcolor = sc.nextLine();
1352     System.out.println("Connecting to the database..."); // Get the database connection, create statement and execute it right away, as no user input need be collected
1353     // try {
1354     //   final Connection connection = DriverManager.getConnection(url);
1355     // } catch (SQLException e) {
1356     //   e.printStackTrace();
1357   } catch (InputMismatchException e) {
1358     System.out.println("Please select one of the options below:");
1359     // Enter new customer;
1360     // Enter a new department;
1361     // Enter a new Process;
1362     // Enter a new Assembly;
1363     // Enter a new Account Number;
1364     // Enter a new Job;
1365     // Enter the completion date of a Job;
1366     // Enter a new transaction Number;
1367     // Retrieve the total cost incurred on an assembly-id;
1368     // Delete all jobs completed on a Department for a given day;
1369     // Retrieve the total labor hours worked on a Department for a given day;
1370     // Retrieve the total labor hours worked on a Department for a given month;
1371     // Retrieve the total labor hours worked on a Department for a given year;
1372     // Retrieve the jobs completed during given date in a given department;
1373     // Retrieve the jobs completed during given month in a given department;
1374     // Retrieve the jobs completed during given year in a given department;
1375     // Delete all cut-jobs with a Job Number;
1376     // Change the color of a Paint Job;
1377     // Import: enter new customers from a data file;
1378     // Export Customer Data;
1379     // Exit!
1380     System.out.println("Please enter the Job Number for the color change:");
1381     final int jobno = sc.nextInt();
1382     final String color = sc.nextLine();
1383     System.out.println("Please enter the new color for the job:");
1384     final String newcolor = sc.nextLine();
1385     System.out.println("Connecting to the database..."); // Dispatching the query 15...
1386     // Dispatching the query 15...
1387     // Done. 1 rows updated in "paintjob" table..
1388   }

```

Result:

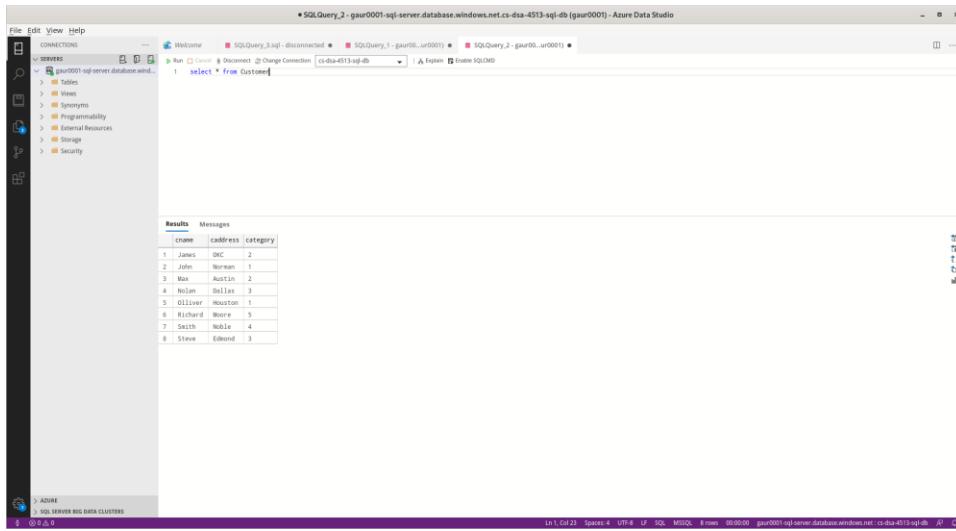
```
File Edit View Help
CONNECTIONS
  Welcome SQLQuery_3 (sql - disconnected) SQLQuery_1 - gaur001_ur0001 SQLQuery_2 - gaur001_ur0001
  Run Disconnect Change Connection cs-dba-4513-sql-db | A Export Create SQLDOD
  SERVERS
  gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db
  Tables
  Views
  Synonyms
  Programmability
  External Resources
  Storage
  Security
  AZURE
  SQL SERVER BIG DATA CLUSTERS
  0, 0, 0
  Results Messages
  job_id color volume labor_time
  1 Carmel 1 1
  2 Lavender 2 1
  3 Bordeaux 4 1
  Line 1, Col 24 Spaces: 4 UTF-8 LF SQL MSQQL 3 rows 00:00:00 gaur001-sql-server.database.windows.net.cs-dba-4513-sql-db
```

One Query for Type 16:

First:

```
eclipse-workspace - AzureSqlArc/AzureSQLDBCJava - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
File Edit Source Refactor Navigate Search Project Run Window Help
src
  package com;
  import java.sql.*;
  public class HelloWorld {
    public static void main(String[] args) {
      String line = "1077,Carmel,1,1";
      String[] customer = line.split(","); // use comma as separator
      System.out.println("Customer Name: " + customer[0] + "Address" + customer[1] + "\nCategory" + customer[2]);
      PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_1);
      //Prepare the query template with the data collected from the user
      statement1.setString(1, customer[0]);
      statement1.setString(2, customer[1]);
      statement1.setInt(3, Integer.parseInt(customer[2]));
      statement1.executeUpdate();
      System.out.println("Dispatching the query ...");
      //Execute the query
      final int rows_inserted = statement1.executeUpdate();
      System.out.println(String.format("Done. %d rows inserted in `customer` table..", rows_inserted));
    }
  }
  Problems in javadoc Declaration
  EclipseSQLDBCJava [Java Application] /usr/lib/eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210528-1205/nbirv/java (Nov 23, 2021, 4:49:54 PM)
  Please select one of the options below:
  1) Enter new customer;
  2) Enter a new Address;
  3) Enter a new Process;
  4) Enter a new Assembly;
  5) Enter a new Product Number;
  6) Enter a new Job;
  7) Enter a new Revision date of a Job;
  8) Enter a new Transaction Number;
  9) Retrieve the total cost incurred on an assembly-id;
  10) Retrieve the total time within a Department for a given day;
  11) Retrieve the process through which a given assembly-id has passed;
  12) Retrieve the revision date of a job within a department;
  13) Retrieve the customers (in name order) whose category is in a given range;
  14) Delete all cut-jobs with a job Number;
  15) Import: enter new customers from a data file;
  16) Import: enter new customers from a data file;
  17) Exit!
  18) Exit!
  Please Enter the path of the file for importing data:
  azure_import.csv
  Dispatching the query 1...
  Done. 1 rows inserted in `customer` table..
  Dispatching the query 1...
  Done. 1 rows inserted in `customer` table..
  Dispatching the query 1...
  Done. 1 rows inserted in `customer` table..
  Please select one of the options below:
```

Result:



Azure Data Studio interface showing a query result for the 'Customers' table. The results are displayed in a table with columns: name, address, and category.

	name	address	category
1	James	DMC	2
2	John	NYC	1
3	Max	Austin	2
4	Nolan	Dallas	3
5	Oliver	Houston	1
6	Richard	Moore	5
7	Smith	Noble	4
8	Steve	Edmond	3

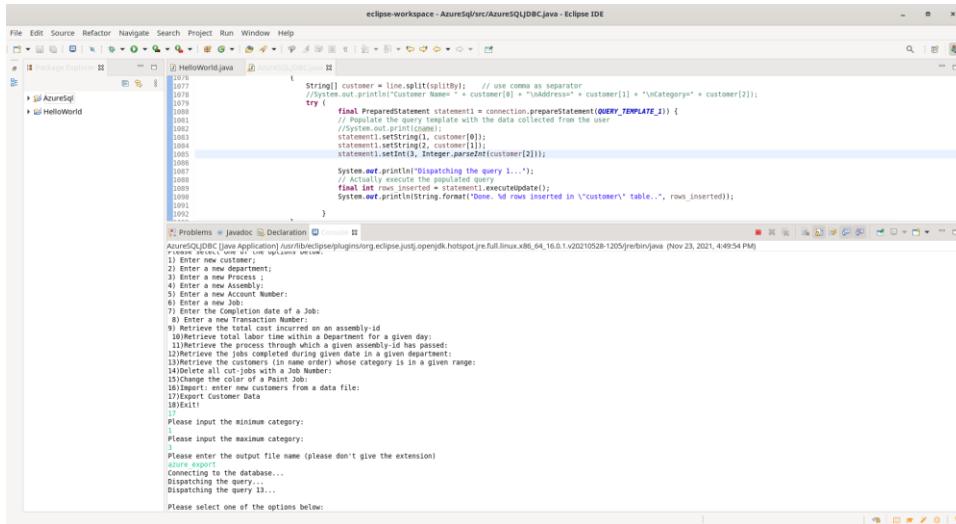
The file:



A screenshot of a CSV file named "azure_import.csv". The file contains the following data:

```
1 Oliver,Houston,1
2 Max,Austin,2
3 Nolan,Dallas,3
```

One Query of Type 17:



```

eclipse-workspace - AzureSql/src/AzureSQLJDBC.java - Eclipse IDE

File Edit Source Refactor Search Project Run Window Help
File Edit Source Refactor Search Project Run Window Help
HelloWorld.java 1 package com;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class HelloWorld {
7     public static void main(String[] args) {
8         String line;
9         String[] customer;
10        try {
11            final PreparedStatement statement1 = connection.prepareStatement(QUERY_TEMPLATE_1);
12            //Populate the statement with the data collected from the user
13            statement1.setString(1, customer[0]);
14            statement1.setString(2, customer[1]);
15            statement1.setInt(3, Integer.parseInt(customer[2]));
16
17            System.out.println("Dispatching the query 1..."); 
18            //Actually execute the populated query
19            final int rows_inserted = statement1.executeUpdate();
20            System.out.println("Done. " + rows_inserted + " rows inserted in `customer` table..");
21        } catch (Exception e) {
22            e.printStackTrace();
23        }
24    }
25}

```

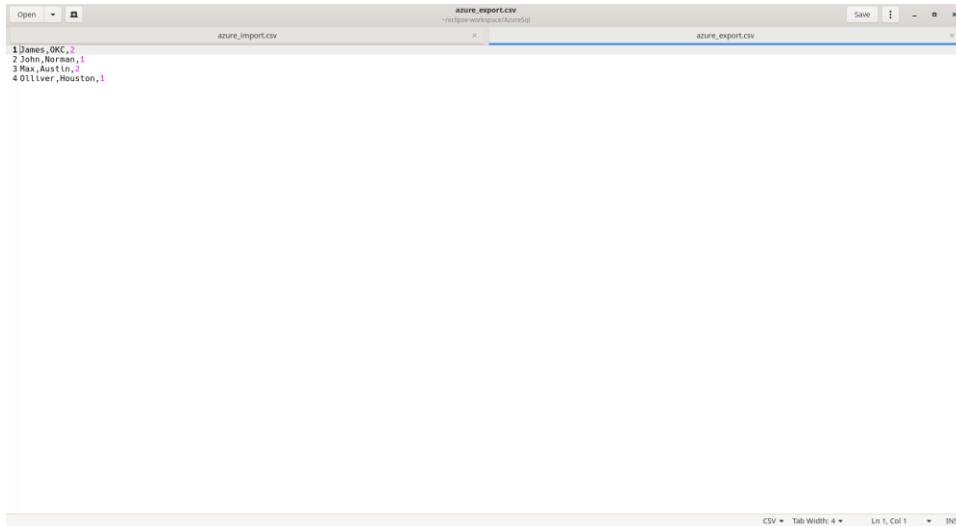
* Problems in Javadoc * Description

AzureSQLJDBC [Java Application] Run/Favorites Help Plugins org.eclipse.justmyjava.openjdk.hotspot.jre.full.linux.x86_64_16.0.1.v20210526-1205/rebir/java (Nov 23, 2021, 4:49:54 PM)

1) Enter a new customer:
2) Enter a new department:
3) Enter a new Process:
4) Enter a new Product:
5) Enter a new Account Number:
6) Enter a new Address:
7) Enter the completion date of a Job:
8) Enter a new Transaction Number:
9) Enter a new Employee assigned on an assembly-line:
10) Retrieve total labor time within a Department for a given day:
11) Retrieve the jobs completed during a given date:
12) Retrieve the jobs completed during given date in a given department:
13) Retrieve the customers (in same order) whose category is in a given range:
14) Change the color of a Paint job:
15) Change the color of a Paint job:
16) Import: enter new customers from a data file:
17) Import Customer Data
18) Exit!

Please input the minimum category:
Please input the maximum category:
Please enter the output file name (please don't give the extension)
azure_export
Connecting to the database...
Disposing the query...
Dispatching the query 1...
Please select one of the options below:

Result:

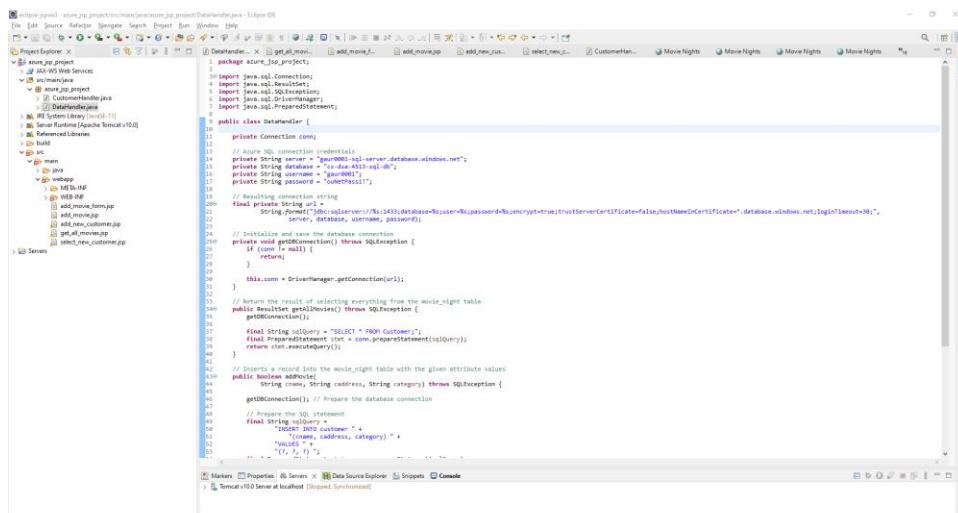


1	Jones,OKC,1		
2	John,Norwalk,1		
3	Max,Austin,2		
4	Diliver,Houston,1		

Task 7. Web Database Application and its execution

7.1 Web Database application source program and screenshots showing its successful compilation

Data Handler



The screenshot shows the Eclipse IDE interface with the DatabaseHandler.java file open in the editor. The code implements a DatabaseHandler class that interacts with an Azure SQL database. It includes methods for getting a connection, executing queries, and preparing statements. The code uses JDBC to connect to the database and execute SQL commands.

```

private Connection conn;
private String server = "yourDBName.sql-server.database.windows.net";
private String database = "yourDBName";
private String username = "yourUserName";
private String password = "yourPassword";
private String url;

public Connection getConnection() throws SQLException {
    String.Format("jdbc:sqlserver://" + url + ";databaseName=" + database + ";user=" + username + ";password=" + password + ";trustServerCertificate=false;hostNameInCertificate=" + database + ".windows.net;loginTimeout=30");
}

private void getDBConnection() throws SQLException {
    conn = DriverManager.getConnection(url);
}

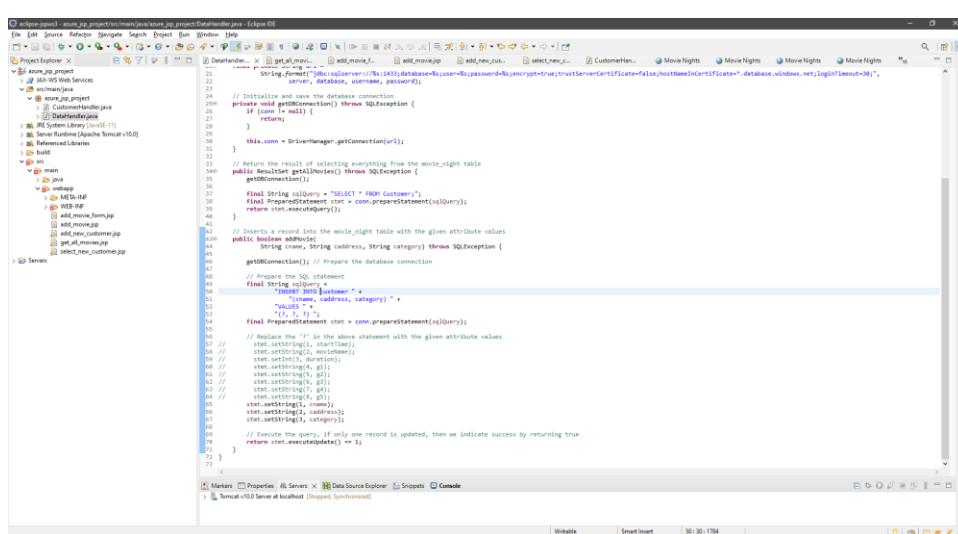
private Statement getStatement() throws SQLException {
    return conn.createStatement();
}

private ResultSet getResults(String query) throws SQLException {
    Statement st = conn.createStatement();
    return st.executeQuery(query);
}

private void addMovie(String name, String address, String category) throws SQLException {
    getDBConnection(); // Prepare the database connection
    final String query = "INSERT INTO customer" +
        "(name, address, category)" +
        "VALUES (%s, %s, %s)";
    final PreparedStatement st = conn.prepareStatement(query);
    st.setString(1, name);
    st.setString(2, address);
    st.setString(3, category);
    st.executeUpdate();
}

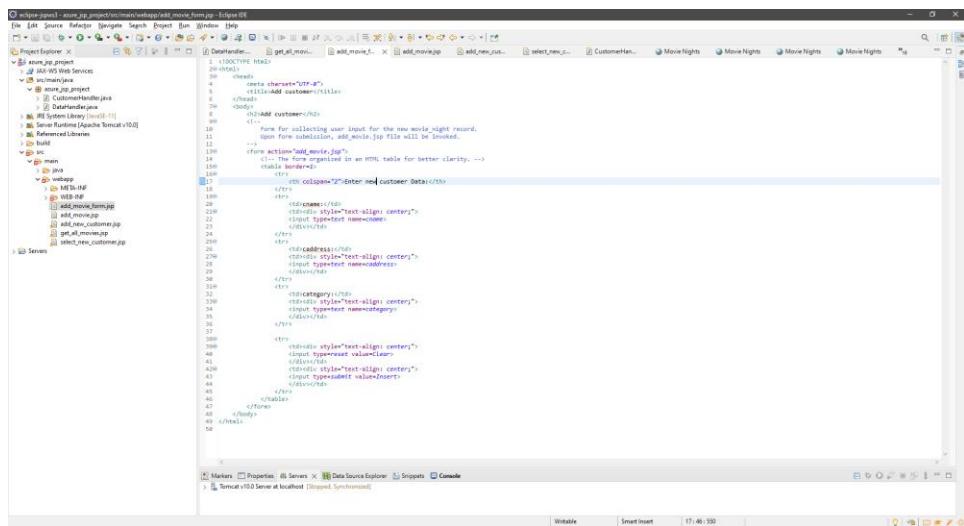
private void addMovieForm(String name, String address, String category) throws SQLException {
    getDBConnection(); // Prepare the database connection
    final String query = "SELECT * FROM customer";
    final PreparedStatement st = conn.prepareStatement(query);
    st.execute();
}

```



The second screenshot shows the same DatabaseHandler.java code in the Eclipse IDE editor. This version includes additional code for handling movie addition and form processing. It uses prepared statements to insert new customers into the database and execute SELECT queries to retrieve results.

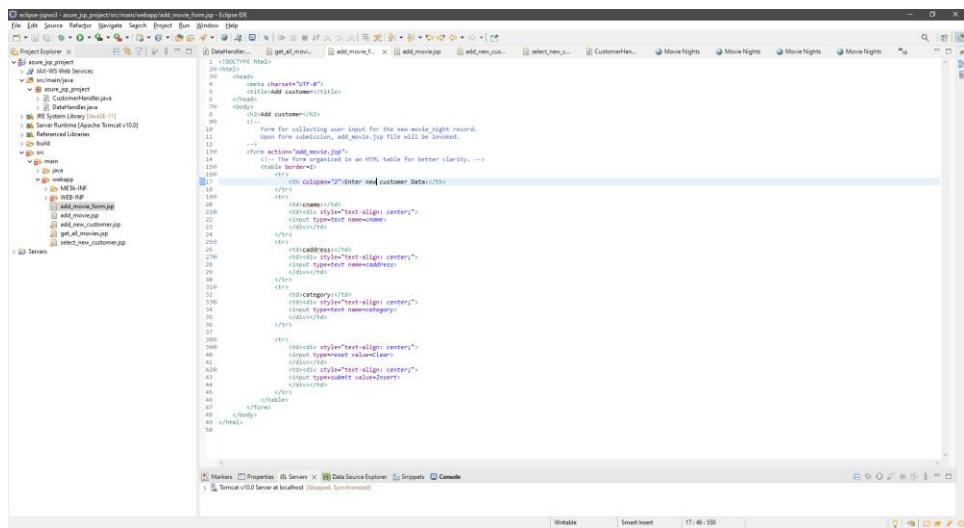
Add_movie_form.jsp (It adds customers, just the name is the same as the file on canvas to reduce complications with building the project)



```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Add customer</title>
        </head>
        <body>
            <h2>Add customer</h2>
            <p>For collecting user input for the new movie_night record.<br/>
                Upon form submission, add_movie.jsp file will be invoked.</p>
            <form action="add_movie.jsp">
                <!-- The form organized in an HTML table for better clarity. -->
                <table border="1">
                    <tr>
                        <td><label>Address</label></td>
                        <td><input type="text" value="123 Main Street" /></td>
                    </tr>
                    <tr>
                        <td><label>City</label></td>
                        <td><input type="text" value="Anytown" /></td>
                    </tr>
                    <tr>
                        <td><label>State</label></td>
                        <td><input type="text" value="CA" /></td>
                    </tr>
                    <tr>
                        <td><label>Zip</label></td>
                        <td><input type="text" value="90210" /></td>
                    </tr>
                    <tr>
                        <td><label>Category</label></td>
                        <td><input type="text" value="1" /></td>
                    </tr>
                    <tr>
                        <td><input type="button" value="Submit" /></td>
                        <td><input type="button" value="Reset" /></td>
                    </tr>
                </table>
            </form>
        </body>
    </html>

```



```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Add customer</title>
        </head>
        <body>
            <h2>Add customer</h2>
            <p>For collecting user input for the new movie_night record.<br/>
                Upon form submission, add_movie.jsp file will be invoked.</p>
            <form action="add_movie.jsp">
                <!-- The form organized in an HTML table for better clarity. -->
                <table border="1">
                    <tr>
                        <td><label>Address</label></td>
                        <td><input type="text" value="123 Main Street" /></td>
                    </tr>
                    <tr>
                        <td><label>City</label></td>
                        <td><input type="text" value="Anytown" /></td>
                    </tr>
                    <tr>
                        <td><label>State</label></td>
                        <td><input type="text" value="CA" /></td>
                    </tr>
                    <tr>
                        <td><label>Zip</label></td>
                        <td><input type="text" value="90210" /></td>
                    </tr>
                    <tr>
                        <td><label>Category</label></td>
                        <td><input type="text" value="1" /></td>
                    </tr>
                    <tr>
                        <td><input type="button" value="Submit" /></td>
                        <td><input type="button" value="Reset" /></td>
                    </tr>
                </table>
            </form>
        </body>
    </html>

```

```

// CustomerHandler.java
1 package com.everis.movieproject;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 public class CustomerHandler extends HttpServlet {
10     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
11         response.getWriter().println("Customer Handler");
12     }
13
14     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15         // The handler is the one in charge of establishing the connection.
16         Databinder handler = new Databinder();
17
18         // Set the attribute values passed from the input form.
19         handler.setAttribute("name", request.getParameter("name"));
20         handler.setAttribute("address", request.getParameter("address"));
21         handler.setAttribute("category", request.getParameter("category"));
22
23         //String duration = request.getParameter("duration");
24         //String guest_1 = request.getParameter("guest_1");
25         //String guest_2 = request.getParameter("guest_2");
26         //String guest_3 = request.getParameter("guest_3");
27         //String guest_4 = request.getParameter("guest_4");
28
29         String course = request.getParameter("course");
30         String category = request.getParameter("category");
31
32         /*
33             * If the user hasn't filled out all the time, movie name and duration. This is very simple checking.
34             */
35         if ((course.equals("")) || address.equals("") || category.equals("")) {
36             response.sendRedirect("add_new_customer.jsp");
37         } else {
38             // Get duration & category parameters from the form.
39             handler.setAttribute("duration", duration);
40             handler.setAttribute("category", category);
41
42             boolean success = handler.addMovie(course, address, category);
43             if (!success) { // Something went wrong
44                 out.println("There was a problem inserting the course!");
45             } else { // Confirm success to the user
46                 out.println("The Customer Added!");
47             }
48
49             out.println("New Customer Successfully Inserted!");
50
51             out.println("See all customers.");
52         }
53     }
54 }

```



```

<%@ page language="java" contentType="text/html; charset=UTF-8"
%>
<!DOCTYPE html PUBLIC "-//IUC//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 Results</title>
</head>
<body>
<%
    Databinder handler = new Databinder();
    handler.imports("com.everis.movieproject.CustomerHandler");
%>
<%
    String course = request.getParameter("course");
    String address = request.getParameter("address");
    String category = request.getParameter("category");
%>
<%
    boolean success = handler.addMovie(course, address, category);
%>
<% if (!success) { %>
<h2>There was a problem inserting the course!</h2>
<% } else { %>
<h2>The Customer Added!</h2>
<% } %>
<%
    out.println("New Customer Successfully Inserted!");
%>
<a href="get_all_movies.jsp">See all customers.</a>
</body>

```



```

<%@ page language="java" contentType="text/html; charset=UTF-8"
%>
<!DOCTYPE html PUBLIC "-//IUC//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta charset="UTF-8">
<title>Add new customer</title>
</head>
<body>
<h2>Add new customer</h2>
<%
    out.print("Now for collecting user input for the new movie night record.");
    out.print("User form submission, add_movie.jsp file will be invoked.");
%>
<form action="add_movie.jsp">
    <table border="1" style="width: 100%; border-collapse: collapse;">
        <tr>
            <td><label>Name:</label></td>
            <td><input type="text" name="name"/></td>
        </tr>
        <tr>
            <td><label>Address:</label></td>
            <td><input type="text" name="address"/></td>
        </tr>
        <tr>
            <td><label>Category:</label></td>
            <td><input type="text" name="category"/></td>
        </tr>
        <tr>
            <td colspan="2" style="text-align: center;">
                <input type="submit" value="Insert" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```



```

<%@ page language="java" contentType="text/html; charset=UTF-8"
%>
<!DOCTYPE html PUBLIC "-//IUC//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta charset="UTF-8">
<title>Customer Database</title>
</head>
<body>
<%
    Databinder handler = new Databinder();
    handler.imports("com.everis.movieproject.CustomerHandler");
    handler.imports("java.util.List");
%>
<%
    final Databinder movies = handler.getMovies();
    final Resultset movies = movies.getResults();
%>
<%
    while(movies.next()) { // For each movie_night record returned...
        // extract the attribute values for every record returned.
        final String name = movies.getAttribute("name");
        final String address = movies.getAttribute("address");
        final String category = movies.getAttribute("category");
        final String guest_1 = movies.getAttribute("guest_1");
        final String guest_2 = movies.getAttribute("guest_2");
        final String guest_3 = movies.getAttribute("guest_3");
        final String guest_4 = movies.getAttribute("guest_4");
        final String duration = movies.getAttribute("duration");
        final String address = movies.getAttribute("address");
        final String category = movies.getAttribute("category");
%>
<%
        out.print("<tr>"); // Start printing out the new table row
        out.print("    <td align='center'>" + name + "</td>");
        out.print("    <td align='center'>" + address + "</td>");
        out.print("    <td align='center'>" + category + "</td>");
        out.print("    <td align='center'>" + duration + "</td>");
        out.print("    <td align='center'>" + guest_1 + "</td>");
        out.print("    <td align='center'>" + guest_2 + "</td>");
        out.print("    <td align='center'>" + guest_3 + "</td>");
        out.print("    <td align='center'>" + guest_4 + "</td>"); 
%>
<%
    }
%>
</table>
</body>
</html>

```

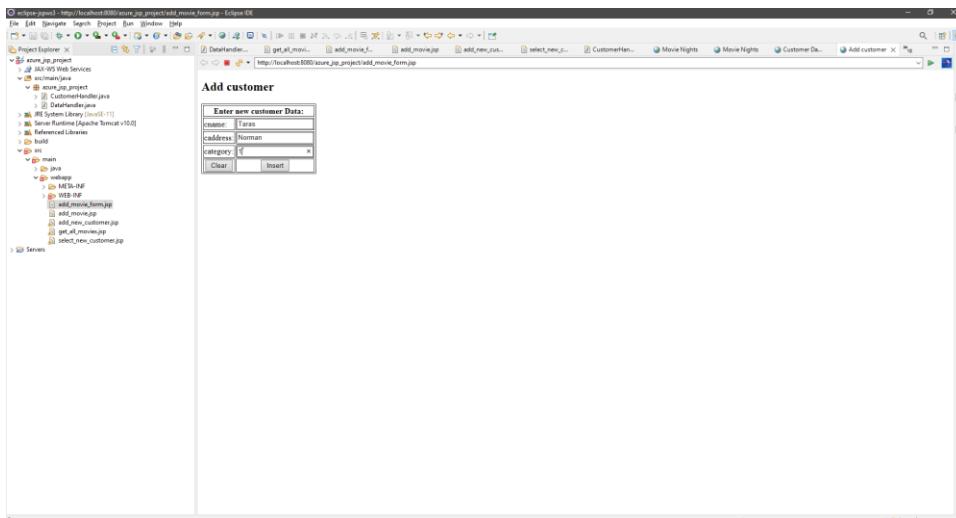
```

<% page language="java" contentType="text/html; charset=UTF-8"
%>
<!DOCTYPE html>
<html>
<head>
<title>Customer Database</title>
</head>
<body>
<table border="1">
<tr>
<th>Name</th>
<th>Start Time</th>
<th>Duration</th>
<th>Guest 1</th>
<th>Guest 2</th>
<th>Address</th>
<th>Category</th>
</tr>
<% ResultSet movies = (ResultSet) request.getAttribute("movies");
final CustomerHandler handler = new CustomerHandler();
while(movies.next()) { // for each movie_night record returned...
    final String name = movies.getString("name");
    final String start_time = movies.getString("start_time");
    final String duration = movies.getString("duration_min");
    final String guest1 = movies.getString("guest_1");
    final String guest2 = movies.getString("guest_2");
    final String address = movies.getString("address");
    final String category = movies.getString("category");
    out.println("<tr>"); // Start printing out the new table row
    out.println("    <td align='center'>" + name +
               "</td><td align='center'>" + start_time +
               "</td><td align='center'>" + duration +
               "</td><td align='center'>" + guest1 +
               "</td><td align='center'>" + guest2 +
               "</td><td align='center'>" + address +
               "</td><td align='center'>" + category +
               "</td></tr>");
}
%>
</table>
<%>
</body>
</html>

```

7.2 Screenshots showing the testing of the Web database application

Adding customers:



The Customer Table:

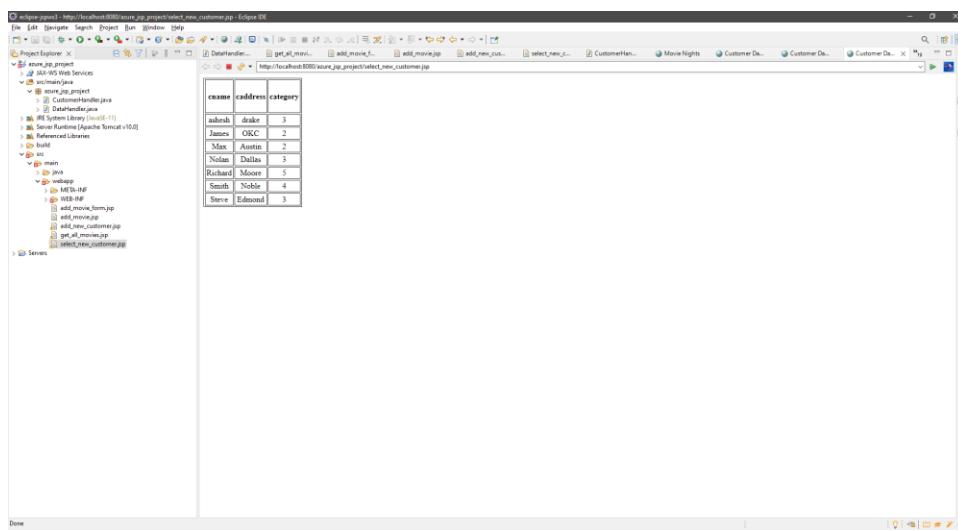
- name: Tara
- address: Norman
- category: 1

New Customer successfully inserted.

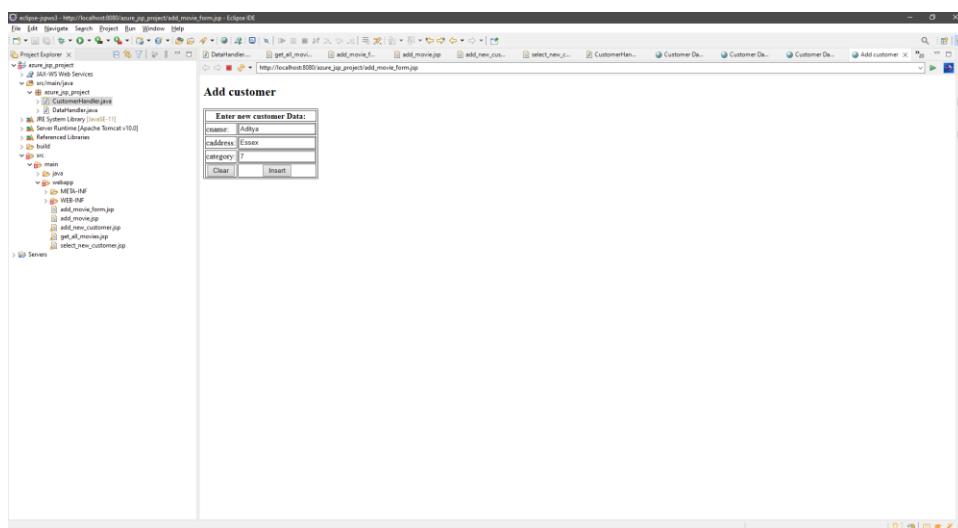
See all Customers.

name	address	category
Abel	Dallas	3
James	OKC	2
John	Norman	1
Max	Austin	2
Nolan	Dallas	3
Oliver	Houston	1
Richard	Moore	5
Smith	Noble	4
Steve	Edmond	3
Tara	Norman	1

Screenshot for Query in Customer range 1 to 5 (Greater than 1 and less than or equal to 5) (Query 13)



Screenshots for Query for adding a new Customer with category 7 (Query 1)



The screenshot shows the Eclipse IDE interface. The Project Explorer on the left lists the 'score_jsp_project' and its contents, including Java files like 'Customer.java', 'CustomerHandler.java', and 'Datamanager.java', and JSP files like 'get_all_movies.jsp', 'add_movie.jsp', 'add_new_customer.jsp', 'select_new_customer.jsp', and 'index.jsp'. The browser window on the right displays the output of a query, showing a table with columns 'cname', 'address', and 'category'. The data in the table is:

cname	address	category
Adriya	Essex	7
ashish	drake	3
James	OKC	2
John	Norman	1
Max	Austin	2
Nolan	Dallas	3
Oliver	Houston	1
Richard	Moore	5
Smith	Noble	4
Steve	Edmond	3
Tara	Norman	1

Screenshot for Query 13 showing Customers within range 1-7 (Greater than 1 and less than equal to 7)

This screenshot is identical to the one above, showing the Eclipse IDE interface with the 'score_jsp_project' selected in the Project Explorer. The browser window displays the same table of customer data, filtered by category values greater than 1 and less than or equal to 7.