

ITD105 Semester 2 2025 – Submission Template

Student Name: Isaac Douglas

Student Number: n12587958

Tutor Name: Mr Mark Walpole



ITD105 Assessment

3B_25s2.sql

MY SCRIPTS:

WARNINGS for Tasks 2-5:

Assignments that do not provide a **detailed and complete** screenshot **of the results** for all tasks or not follow the submission instruction for a given query **will be** awarded Zero.

The screenshot needs to include the **SQL codes, result grid** and **action output**.

Even if your results are in the SQL file/other attached file, your grade will remain zero. We need to follow the same **rule** for all students. So, please make sure to include a **screenshot of the result/s** as requested.

Any submission that attempts to provide a screenshot that does not result from the provided query will be awarded Zero.

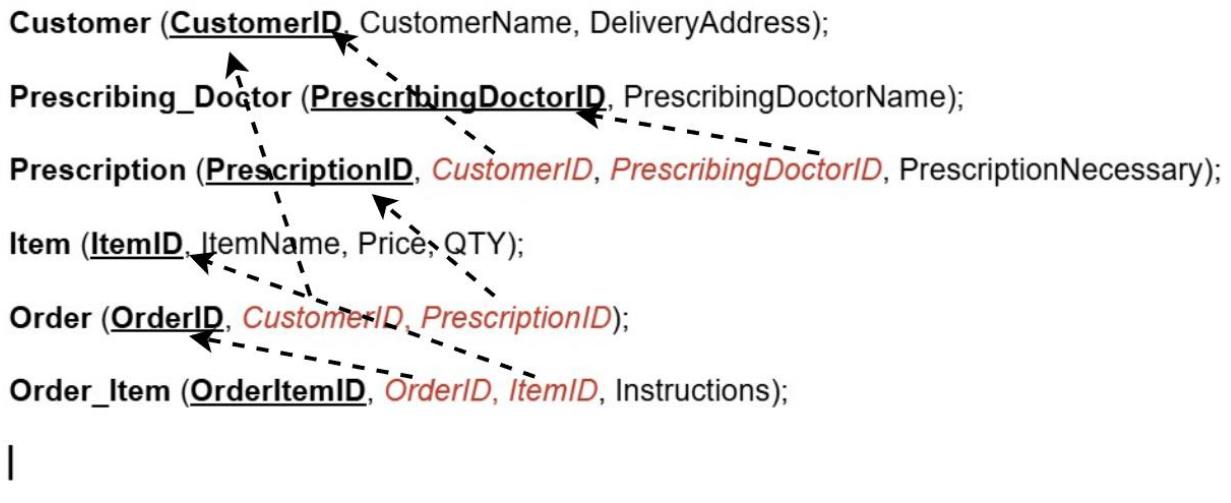
Even, if you are getting **Error message**, you can submit the screenshot of the error message and possible reason. You **might** still get some mark for it!

You can make a screenshot in the following way: Windows <Ctrl + Win + S >, Mac <Shift + Command +4>. You can also use “Snipping Tool” to create screenshots.

The use of artificial intelligence tools (for example Chat GPT) is not permitted in any assessment in this unit. The use of such tools when not authorised may be treated as a breach of MOPP C/5.3 Academic integrity and appropriate penalties imposed.

Task 1 [4 marks]

<insert RMap>



Task 2 [6 marks]

In addition to the screenshots (for all parts), include a SQL script in the file named "ITD105_3B_your_student_number.sql" with your submission. Make sure it clearly shows your script for Task 2.

Please note:

In the following tasks you will be using MySQL Workbench and any question that requires any SQL you must **include a screenshot** of the result from executing your query.

The screenshot needs to include the **SQL codes, result grid and action output** as per the example that follows.

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the following SQL code:

```
1 • USE Assessment_3B_23s2_Dreamhome;
2
3 • SELECT staffNo, fName, lName, Position, mobTelNo, branchNo FROM Staff;
```
- Result Grid:** Displays the results of the query:

staffNo	fName	lName	Position	mobTelNo	branchNo
SN007	Janet	Doe	Teller	0141481825	BN003
SN008	Neal	Walker	Teller	0141881825	BN015
SN009	Mandy	Johnston	Assistant	0141881825	BN006
SN014	Terry	Longbody	Assistant	0141481825	BN003
SN020	Marco	Peters	Teller	0141846825	BN012
SN037	Lia	Smith	Manager	0141881825	BN003
ST005	Shaza	Randall	Assistant	0141841825	BN003
ST021	John	Brown	Assistant	0141841825	BN006
ST041	Jules	Lee	Manager	0141846825	BN009
- Action Output:** Shows the history of actions taken:

#	Time	Action	Message
1	11:42:46	USE Assessment_3B_23s2_Dreamhome	0 row(s) affected
2	11:42:57	SELECT staffNo, fName, lName, Position, mobTelNo, branchNo FROM Staff	9 row(s) returned

MINE:

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Find Done

```

181 -- PropertyListedForSale table
182 CREATE TABLE PropertyListedForSale (
183     propertyNo CHAR(5) NOT NULL,
184     address VARCHAR(50) NOT NULL,
185     postcode VARCHAR(5) NOT NULL,
186     propertyType VARCHAR(15) NOT NULL,
187     rooms SMALLINT NOT NULL,
188     listingPrice DECIMAL(12,2) NOT NULL CHECK (listingPrice > 600000.00),
189     ownerNo CHAR(5) NOT NULL,
190     staffNo CHAR(5) NOT NULL,
191     branchNo CHAR(5) NOT NULL,
192     PRIMARY KEY (propertyNo),
193     -- Foreign keys
194     CONSTRAINT fk_sale_owner FOREIGN KEY (ownerNo) REFERENCES PropertyOwner(ownerNo),
195     CONSTRAINT fk_sale_staff FOREIGN KEY (staffNo) REFERENCES Staff(staffNo),
196     CONSTRAINT fk_sale_branch FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
197 );
198
199
200 -- BrisAuction table

```

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

propertyNo	address	postcode	propertyType	rooms	listingPrice	ownerNo	staffNo	branchNo
PS018	51 South Street New Market	Q4235	House	4	960000.00	P0046	ST021	BN006
PS020	87 King Road Albion	Q4532	House	5	1200000.00	P0076	SN009	BN003
PS064	397 Richmond Kelvin Grove	Q4333	Flat	3	750000.00	P0076	SN007	BN006

PropertyListedForSale 6 x BrisAuction 7 BrisAuctionedProperty 8

Action Output

#	Time	Action	Message	Duration / Fetch
380	12:05:48	INSERT INTO PropertyListedForSale VALUES ('PS020','87 King Road Albion','Q4532','House',5,1200000,P0076,'SN009','BN003')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.00 sec
381	12:05:48	INSERT INTO BrisAuction VALUES ('AN008','2023-01-24,NULL,BN015'), ('AN060','2023-01-04','Auction house 164 P...')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
382	12:05:48	INSERT INTO BrisAuctionedProperty VALUES ('AN008','PS018',950000,950500), ('AN060','PS064',NULL,750000)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

SQL script saved to 'C:\Users\strep\Desktop\AQUIT\ITD105_25CTP2\Database Management\ASS3B\ITD105 Assessment 3B_25s2.sql'

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Find Done

```

199
200 -- BrisAuction table
201 CREATE TABLE BrisAuction (
202     auctionNo CHAR(5) NOT NULL,
203     date DATE NOT NULL,
204     auctionAddress VARCHAR(50) DEFAULT NULL,
205     branchNo CHAR(5) NOT NULL,
206     PRIMARY KEY (auctionNo),
207     -- Foreign key
208     CONSTRAINT fk_auction_branch FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
209 );
210
211 -- BrisAuctionedProperty table
212 CREATE TABLE BrisAuctionedProperty (
213     auctionNo CHAR(5) NOT NULL,
214     propertyNo CHAR(5) NOT NULL,
215     finalBidAmount DECIMAL(12,2) DEFAULT NULL,
216     soldFor DECIMAL(12,2) DEFAULT 600000.00, /*Constraint for "soldFor is $600 000.00 by default.*/
217     PRIMARY KEY (auctionNo, propertyNo),
218     -- Foreign keys

```

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

auctionNo	date	auctionAddress	branchNo
AN008	2023-01-24		BN015
AN060	2023-01-04	Auction house 164 Pain St g Q4051	BN006
AU089	2023-02-13		BN012

PropertyListedForSale 6 BrisAuction 7 BrisAuctionedProperty 8

Action Output

#	Time	Action	Message	Duration / Fetch
380	12:05:48	INSERT INTO PropertyListedForSale VALUES ('PS020','87 King Road Albion','Q4532','House',5,1200000,P0076,'SN009','BN003')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.00 sec
381	12:05:48	INSERT INTO BrisAuction VALUES ('AN008','2023-01-24,NULL,BN015'), ('AN060','2023-01-04','Auction house 164 P...')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
382	12:05:48	INSERT INTO BrisAuctionedProperty VALUES ('AN008','PS018',950000,950500), ('AN060','PS064',NULL,750000)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

SQL script saved to 'C:\Users\strep\Desktop\AQUIT\ITD105_25CTP2\Database Management\ASS3B\ITD105 Assessment 3B_25s2.sql'

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Find Done

```
208     CONSTRAINT fk_auction_branch FOREIGN KEY (branchNo) REFERENCES Branch(branchNo)
209   );
210
211   -- BrisAuctionedProperty table
212   CREATE TABLE BrisAuctionedProperty (
213     auctionNo CHAR(9) NOT NULL,
214     propertyNo CHAR(9) NOT NULL,
215     finalBidAmount DECIMAL(12,2) DEFAULT NULL,
216     soldFor DECIMAL(12,2) DEFAULT 600000.00, #Constraint for "soldFor" is $600 000.00 by default.
217     PRIMARY KEY (auctionNo, propertyNo),
218     -- Foreign keys
219     CONSTRAINT fk_auctioned_auction FOREIGN KEY (auctionNo) REFERENCES BrisAuction(auctionNo),
220     CONSTRAINT fk_auctioned_property FOREIGN KEY (propertyNo) REFERENCES PropertyListedForSale(propertyNo)
221   );
222
223   -- Sample INSERTs based on your data
224
225   ● INSERT INTO PropertyListedForSale VALUES
226     ('PS020','87 King Road Albion','Q4532','House',5,1200000,`PO076`,`SN095`,`BN003`),
227     ('PS018','51 South Street New Market','Q4235','House',4,960000,`PO040`,`ST021`,`BN005`),
228     ('PS064','397 Richmond Kelvin Grove','Q4333','Flat',3,750000,`PO070`,`SN087`,`SN005`);
229
230   -- Sample INSERTs based on your data
231
232   ● SELECT * FROM PropertyListedForSale;
233   ● SELECT * FROM BrisAuction;
```

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

auctionNo	propertyNo	finalBidAmount	soldFor
AN008	PS018	950000.00	950500.00
AN060	PS064	NULL	750000.00
AN008	PS018	NULL	NULL

PropertyListedForSale 6 BrisAuction 7 BrisAuctionedProperty 8 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
380	12:05:48	INSERT INTO PropertyListedForSale VALUES ('PS020','87 King Road Albion','Q4532','House',5,1200000,`PO076`,`SN095`,`BN003')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
381	12:05:48	INSERT INTO BrisAuction VALUES ('AN008','2023-01-24',NULL,BN015), ('AN060','2023-01-04','Auction house 164 Pain St g Q4051','BN006')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
382	12:05:48	INSERT INTO BrisAuctionedProperty VALUES ('AN008','PS018',950000,950500), ('AN060','PS064',NULL,750000)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

SQL script saved to 'C:\Users\strep\Desktop\AUQT\TD105_25CTP2 Database Management\ASS3B\TD105 Assessment 3B_25z2.sql'

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Find Done

```
223   -- Sample INSERTs based on your data
224
225   ● INSERT INTO PropertyListedForSale VALUES
226     ('PS020','87 King Road Albion','Q4532','House',5,1200000,`PO076`,`SN095`,`BN003`),
227     ('PS018','51 South Street New Market','Q4235','House',4,960000,`PO040`,`ST021`,`BN005`),
228     ('PS064','397 Richmond Kelvin Grove','Q4333','Flat',3,750000,`PO070`,`SN087`,`SN005`);
229
230   ● INSERT INTO BrisAuction VALUES
231     ('AN008','2023-01-24',NULL,BN15),
232     ('AN060','2023-01-04','Auction house 164 Pain St g Q4051','BN006'),
233     ('AN008','2023-02-13',NULL,BN12);
234
235   ● INSERT INTO BrisAuctionedProperty VALUES
236     ('AN008','PS018',950000,950500),
237     ('AN060','PS064',NULL,750000);
238
239
240
241   ● SELECT * FROM PropertyListedForSale;
242   ● SELECT * FROM BrisAuction;
```

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

auctionNo	propertyNo	finalBidAmount	soldFor
AN008	PS018	950000.00	950500.00
AN060	PS064	NULL	750000.00
AN008	PS018	NULL	NULL

PropertyListedForSale 6 BrisAuction 7 BrisAuctionedProperty 8 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
380	12:05:48	INSERT INTO PropertyListedForSale VALUES ('PS020','87 King Road Albion','Q4532','House',5,1200000,`PO076`,`SN095`,`BN003')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
381	12:05:48	INSERT INTO BrisAuction VALUES ('AN008','2023-01-24',NULL,BN015), ('AN060','2023-01-04','Auction house 164 Pain St g Q4051','BN006')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
382	12:05:48	INSERT INTO BrisAuctionedProperty VALUES ('AN008','PS018',950000,950500), ('AN060','PS064',NULL,750000)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.000 sec

SQL script saved to 'C:\Users\strep\Desktop\AUQT\TD105_25CTP2 Database Management\ASS3B\TD105 Assessment 3B_25z2.sql'

Task 3 [6 marks]

For each section:

<insert SQL statement>

```
SELECT clientNo, fName, lName, emailAddr, prefType, maxRent
```

FROM Client

```
WHERE prefType = 'Flat'
```

AND maxRent < 560;

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

JD105 Assessment 3B_25e2* ×

Find Done

```

239 • SELECT * FROM PropertyListedForSale;
240 • SELECT * FROM BrisAuction;
241 • SELECT * FROM BrisAuctionedProperty;
242
243
244 • SELECT clientNo, fName, lName, emailAddr, prefType, maxRent
  FROM Client
  WHERE prefType = 'Flat'
    AND maxRent < 560;
245
246 • INSERT INTO Client (clientNo, fName, lName, address, mobTelNo, emailAddr, prefType, maxRent)
  VALUES ('CR87', 'Marry', 'Smith', '7 Shark St Brisbane city', '0141003590', 'smith@hotmail.com.au', 'Flat', 640);
251
252 • DELETE FROM Inspection
  WHERE clientNo = 'CN75'
    AND propertyNo = 'PF016'
      AND viewDate = '2023-04-28';
254

```

In fo

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

clientNo	fName	lName	emailAddr	prefType	maxRent
CN65	Robert	De Niro	rdenero@hotmail.com.au	Flat	470.0
CN85	Tom	Hanks	thanks@hotmail.com.au	Flat	460.0
NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor

PropertyListedForSale6 Client 10 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
387	13:12:46	INSERT INTO Client (clientNo, fName, lName, address, mobTelNo, emailAddr, prefType, maxRent) VALUES ('CR87', ...)	1 row(s) affected	0.032 sec
388	13:14:05	SELECT clientNo, fName, lName, emailAddr, prefType, maxRent FROM Client WHERE prefType = 'Flat' AND maxRent...	2 row(s) returned	0.000 sec / 0.000 sec

```

INSERT INTO Client (clientNo, fName, lName, address, mobTelNo, emailAddr, prefType, maxRent)
VALUES ('CR87', 'Marry', 'Smith', '7 Shark St Brisbane city', '0141003590',
'smith@hotmail.com.au', 'Flat', 640);

```

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

JD105 Assessment 3B_25e2* ×

Find Done

```

239 • SELECT * FROM PropertyListedForSale;
240 • SELECT * FROM BrisAuction;
241 • SELECT * FROM BrisAuctionedProperty;
242
243
244 • SELECT clientNo, fName, lName, emailAddr, prefType, maxRent
  FROM Client
  WHERE prefType = 'Flat'
    AND maxRent < 560;
245
246 • INSERT INTO Client (clientNo, fName, lName, address, mobTelNo, emailAddr, prefType, maxRent)
  VALUES ('CR87', 'Marry', 'Smith', '7 Shark St Brisbane city', '0141003590', 'smith@hotmail.com.au', 'Flat', 640);
251
252 • DELETE FROM Inspection
  WHERE clientNo = 'CN75'
    AND propertyNo = 'PF016'
      AND viewDate = '2023-04-28';
254

```

In fo

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

propertyNo	address	postcode	propertyType	rooms	listingPrice	ownerNo	staffNo	brandNo
PS018	51 South Street New Market	Q4235	House	4	960000.00	P0046	ST021	BN006
PS020	87 King Road Albion	Q4532	House	5	1200000.00	P0076	SN009	BN003
PS064	397 Richmond Kelvin Grove	Q4333	Flat	3	750000.00	P0075	SN007	BN006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor

PropertyListedForSale6 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
388	13:14:05	SELECT clientNo, fName, lName, emailAddr, prefType, maxRent FROM Client WHERE prefType = 'Flat' AND maxRent...	2 row(s) returned	0.000 sec / 0.000 sec
389	13:14:17	INSERT INTO Client (clientNo, fName, lName, address, mobTelNo, emailAddr, prefType, maxRent) VALUES ('CR87', ...)	Error Code: 1062: Duplicate entry 'CR87' for key 'client.PRIMARY'	0.000 sec

```
DELETE FROM Inspection
```

```
WHERE clientNo = 'CN75'
```

```
AND propertyNo = 'PF016'
```

```
AND viewDate = '2023-04-20';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the DELETE query with the line `DELETE FROM Inspection WHERE clientNo = 'CN75' AND propertyNo = 'PF016' AND viewDate = '2023-04-20';` highlighted.
- Result Grid:** Shows a table of property listings. One row is selected, corresponding to the deleted record.
- Action Output:** Logs two events:
 - DELETE FROM Inspection WHERE clientNo = 'CN75' AND propertyNo = 'PF016' AND viewDate = '2023-04-20' - 1 row(s) affected (Duration: 0.047 sec)
 - DELETE FROM Inspection WHERE clientNo = 'CN75' AND propertyNo = 'PF016' AND viewDate = '2023-04-20' - 0 row(s) affected (Duration: 0.000 sec)
- Output:** Shows the message "Query Completed".

```
UPDATE Staff
```

```
SET salary = salary * 1.24
```

```
WHERE staffNo IN ('SN037', 'ST041');
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the UPDATE query with the line `SET salary = salary * 1.24 WHERE staffNo IN ('SN037', 'ST041');` highlighted.
- Result Grid:** Shows a table of property listings. The same row from the previous screenshot is selected.
- Action Output:** Logs two events:
 - DELETE FROM Inspection WHERE clientNo = 'CN75' AND propertyNo = 'PF016' AND viewDate = '2023-04-20' - 0 row(s) affected (Duration: 0.000 sec)
 - UPDATE Staff SET salary = salary * 1.24 WHERE staffNo IN ('SN037', 'ST041') - 2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0 (Duration: 0.031 sec)
- Output:** Shows the message "Query Completed".

```

SELECT COUNT(*) AS num_properties #count all rows for P0046 in PropertyForRent and display result
in num_properties column

FROM PropertyForRent

WHERE ownerNo = 'P0046';

```

```

260
261 • SELECT COUNT(*) AS num_properties
262   FROM PropertyForRent
263   WHERE ownerNo = 'P0046';
264
265 • SELECT *
266   FROM PropertyForRent
267   WHERE address LIKE '2%'
268     OR address LIKE '5%';
269
270 • SELECT b.address AS branchAddress, COUNT(p.propertyNo) AS totalProperties
271   FROM Branch b
272   JOIN PropertyForRent p ON b.branchNo = p.branchNo
273   GROUP BY b.branchNo, b.address
274   HAVING COUNT(p.propertyNo) > 1;
275

```

Result Grid	
	num_properties
▶	2

PropertyListedForSale6 Result 11 x

Action Output

#	Time	Action	Message	Duration / Fetch
392	13:15:10	UPDATE Staff SET salary = salary * 1.24 WHERE staffNo IN ('SN037', 'ST041')	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0	0.031 sec
393	13:15:25	SELECT COUNT(*) AS num_properties FROM PropertyForRent WHERE ownerNo = 'P0046'	1 row(s) returned	0.000 sec / 0.000 sec

```

SELECT *

FROM PropertyForRent

```

WHERE address LIKE '2%' # matches any sequence of characters after the starting character and
LIKE is a matching operator

OR address LIKE '5%';

```

254     AND propertyNo = 'PF016'
255     AND viewDate = '2023-04-20';
256
257 • UPDATE Staff
258   SET salary = salary * 1.24
259   WHERE staffNo IN ('SN037', 'ST041');
260
261 • SELECT COUNT(*) AS num_properties
262   FROM PropertyForRent
263   WHERE ownerNo = 'P0046';
264
265 • SELECT *
266   FROM PropertyForRent
267   WHERE address LIKE '2%'
268     OR address LIKE '5%';
269
270 • SELECT b.address AS branchAddress, COUNT(p.propertyNo) AS totalProperties

```

Result Grid	
	num_properties
▶	2

PropertyListedForSale6 Result 11 x

Action Output

#	Time	Action	Message	Duration / Fetch
392	13:15:10	UPDATE Staff SET salary = salary * 1.24 WHERE staffNo IN ('SN037', 'ST041')	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0	0.031 sec
393	13:15:25	SELECT COUNT(*) AS num_properties FROM PropertyForRent WHERE ownerNo = 'P0046'	1 row(s) returned	0.000 sec / 0.000 sec

```

SELECT b.address AS branchAddress, COUNT(p.propertyNo) AS totalProperties
FROM Branch b #finds branches
JOIN PropertyForRent p ON b.branchNo = p.branchNo #joins branch to its properties
GROUP BY b.branchNo, b.address
HAVING COUNT(p.propertyNo) > 1; #displays properties over 1

```

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the provided SQL query. The results grid shows two rows of data:

branchAddress	totalProperties
56 Dover Drive Williston	2
153 Pain Street Enoggera	2

The action output tab shows the following log entries:

#	Time	Action	Message	Duration / Fetch
393	13:15:25	SELECT COUNT(*) AS num_properties FROM PropertyForRent WHERE ownerNo = 'P0046'	1 row(s) returned	0.000 sec / 0.000 sec
394	13:15:42	SELECT b.address AS branchAddress, COUNT(p.propertyNo) AS totalProperties FROM Branch b JOIN PropertyForRe...	2 row(s) returned	0.016 sec / 0.000 sec

<insert screenshot as requested >

The screenshot (for all parts) needs to include the **SQL codes**, **result grid** and **action output**.

Task 4 [7 marks]

For each section:

<insert SQL statement>

```

SELECT *
FROM Client
WHERE clientNo IN (
    SELECT clientNo
    FROM Inspection
    WHERE viewDate < '2023-05-15'
);

```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

ITD105 Assessment 3B_25e2-1* ITD105 Assessment 3B_25e2-1*

```

169 • INSERT INTO Registration VALUES ('CN85', 'BN009', 'ST041', '2021-03-07');
170
171 -- SELECT * FROM Registration;
172
173 -- End of script
174
175 #Task 4
176 • SELECT *
177 FROM Client
178 WHERE clientNo IN (
179     SELECT clientNo
180     FROM Inspection
181     WHERE viewDate < '2023-05-15'
182 );
183
184 • SELECT DISTINCT po.fName, po.lName, po.address
185     FROM PropertyOwner po

```

Result Grid

clientNo	fName	lName	address	mobTelNo	emailAddr	prefType	maxRent
CN65	Katherine	Hepburn	5 AfricanQueen St Congo Q 4568	0141881825	khepburn@hotmail.com.au	Flat	560.0
CN75	Humphrey	Bogart	5 Casablanca St RidsPlace Q 4569	0141881825	hbogart@hotmail.com.au	House	830.0
CN85	Tom	Hanks	5 Castaway St LonelyIsland Q 4566	0141841825	thanks@hotmail.com.au	Flat	460.0

Client 1

Output

Action Output

#	Time	Action	Message	Duration / Fetch
443	13:36:36	INSERT INTO Registration VALUES ('CN85', 'BN009', 'ST041', '2021-03-07')	1 row(s) affected	0.000 sec
444	13:40:11	SELECT * FROM Client WHERE clientNo IN (SELECT clientNo FROM Inspection WHERE viewDate < '2023-05-15')	3 row(s) returned	0.000 sec / 0.000 sec

SELECT DISTINCT po.fName, po.lName, po.address #DISTINCT ensures each owner appears only once if they own multiple properties.

FROM PropertyOwner po #define the alias we used before "po"

JOIN PropertyForRent pfr ON po.ownerNo = pfr.ownerNo #defined alias pfr adn creates a result set

WHERE pfr.rooms > 3;

File Edit View Query Database Server Tools Scripting Help

ITD105 Assessment 3B_25e2-1* ITD105 Assessment 3B_25e2-1*

```

179 WHERE clientNo IN (
180     SELECT clientNo
181     FROM Inspection
182     WHERE viewDate < '2023-05-15'
183 );
184
185 • SELECT DISTINCT po.fName, po.lName, po.address #DISTINCT ensures each owner appears only once if they own multiple properties.
186     FROM PropertyOwner po #define the alias we used before "po"
187     JOIN PropertyForRent pfr ON po.ownerNo = pfr.ownerNo #defined alias pfr
188     WHERE pfr.rooms > 3;
189
190 • SELECT DISTINCT c.fName, c.lName
191     FROM Client c
192     JOIN Inspection i ON c.clientNo = i.clientNo
193     JOIN PropertyForRent pfr ON i.propertyNo = pfr.propertyNo
194     JOIN Branch b ON pfr.branchNo = b.branchNo

```

Result Grid

fName	lName	address
Ike	Turner	18 Well St, Windsor Q4030
Peta	Sims	20 ParkPl, Windsor Q4030
Maggie	Simpson	16 Adriay St, Windsor Q4030

Result 3

Output

Action Output

#	Time	Action	Message	Duration / Fetch
447	13:42:57	SELECT * from Inspection	5 row(s) returned	0.000 sec / 0.000 sec
448	13:46:25	SELECT DISTINCT po.fName, po.lName, po.address #DISTINCT ensures each owner appears only once if they own ...	3 row(s) returned	0.000 sec / 0.000 sec

SELECT DISTINCT c.fName, c.lName

FROM Client c

JOIN Inspection i ON c.clientNo = i.clientNo #match client and inspection

```

JOIN PropertyForRent pfr ON i.propertyNo = pfr.propertyNo #match inspection to the property
JOIN Branch b ON pfr.branchNo = b.branchNo #match property to branch that manages it. All run in sequence which affect the results
WHERE b.postcode = 'Q4044'; #UI

```

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Local instance MySQL80 x
ITD105 Assessment 3B_25a2-1*
Find C:\po\Name Done
183
184
185 • SELECT DISTINCT po.fName, po.lName, po.address #DISTINCT ensures each owner appears only once if they own multiple properties.
186     FROM PropertyOwner po #define the alias we used before "po"
187     JOIN PropertyForRent pfr ON po.ownerNo = pfr.ownerNo #defined alias pfr adn creates a result set
188     WHERE pfr.rooms > 3;
189
190 • SELECT DISTINCT c.fName, c.lName
191     FROM Client c
192     JOIN Inspection i ON c.clientNo = i.clientNo #match client and inspection
193     JOIN PropertyForRent pfr ON i.propertyNo = pfr.propertyNo #match inspection to the property
194     JOIN Branch b ON pfr.branchNo = b.branchNo #match property to branch that manages it. All run in sequence which affect the results
195     WHERE b.postcode = 'Q4044'; #UI
196
197 • SELECT b.address, b.postcode,
198     SUM(DISTINCT s.salary) AS totalSalary #Adds total amount of each salary with no duplicates because of Distinct

```

The result grid shows a single row:

Name	lName
Katherine	Hepburn

The action output pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
451	14:07:41	SELECT b.address, b.postcode, SUM(DISTINCT s.salary) AS totalSalary; #Adds total amount of each salary with no duplicates because of Distinct	5 row(s) returned	0.015 sec / 0.000 sec
452	14:11:38	SELECT DISTINCT c.fName, c.lName FROM Client c JOIN Inspection i ON c.clientNo = i.clientNo #match client and i...	1 row(s) returned	0.000 sec / 0.000 sec

Query Completed

```

SELECT b.address, b.postcode,
       SUM(DISTINCT s.salary) AS totalSalary, #Adds total amount of each salary with no duplicates because of Distinct
       COUNT(DISTINCT s.staffNo) AS numEmployees, #Dont want to add staffNo, instead count rows of StaffNo's and properties
       COUNT(DISTINCT p.propertyNo) AS numProperties
FROM Branch b #displays
LEFT JOIN Staff s ON b.branchNo = s.branchNo #ensures branches with no staff or properties are included
LEFT JOIN PropertyForRent p ON b.branchNo = p.branchNo #almost like the constraints
GROUP BY b.branchNo, b.address, b.postcode;

```

The screenshot shows the MySQL Workbench interface with an SQL query editor and a results grid.

```

191 FROM Client c
192 JOIN Inspection i ON c.clientNo = i.clientNo #match client and inspection
193 JOIN PropertyForRent pfr ON i.propertyNo = pfr.propertyNo #match inspection to the property
194 JOIN Branch b ON pfr.branchNo = b.branchNo #match property to branch that manages it. All run in sequence which affect the results
195 WHERE b.postcode = 'Q4044' #UI
196
197 • SELECT b.address, b.postcode,
198     SUM(DISTINCT s.salary) AS totalSalary, #Adds total amount of each salary with no duplicates because of Distinct
199     COUNT(DISTINCT s.staffNo) AS numEmployees, #Dont want to add staffNo, instead count rows of StaffNo's and properties
200     COUNT(DISTINCT p.propertyNo) AS numProperties
201 FROM Branch b #displays
202 LEFT JOIN Staff s ON b.branchNo = s.branchNo #ensures branches with no staff or properties are included
203 LEFT JOIN PropertyForRent p ON b.branchNo = p.branchNo #almost like the constraints
204 GROUP BY b.branchNo, b.address, b.postcode
205

```

Result Grid:

address	postcode	totalSalary	numEmployees	numProperties
56 Dover Drive Wilston	Q4041	376000.00	4	2
163 Pain Street Enoggera	Q4041	176000.00	2	2
32 Chance Road Zillmere	Q4043	105000.00	1	1
22 Steer Road East Brisbane	Q4044	85000.00	1	1
16 Argon Street Cleveland	Q4045	81000.00	1	1

Action Output:

Action	Time	Message	Duration / Fetch
450	13:56:28	SELECT b.address, b.postcode, SUM(DISTINCT s.salary) AS totalSalary, COUNT(DISTINCT s.staffNo) AS numEmployees, COUNT(DISTINCT p.propertyNo) AS numProperties	0.031 sec / 0.000 sec
451	14:07:41	SELECT b.address, b.postcode, SUM(DISTINCT s.salary) AS totalSalary, #Adds total amount of each salary with no duplicates because of Distinct COUNT(DISTINCT s.staffNo) AS numEmployees, COUNT(DISTINCT p.propertyNo) AS numProperties	0.015 sec / 0.000 sec

<insert screenshot as requested >

The screenshot (for all parts) needs to include the **SQL codes**, **result grid** and **action output**.

Task 5 [7 marks]

In addition to the screenshots (for all parts), include a SQL script in the file named "ITD105_3B_your_student_number.sql" with your submission. Make sure it clearly shows Task 5.

Please note:

In the following tasks you will be using MySQL Workbench and any question that requires any SQL you must **include a screenshot** of the result from executing your query.

The screenshot needs to include the **SQL codes**, **result grid** and **action output** as per the example that follows.

The screenshot shows the MySQL Workbench interface with a query editor window. The code being run is:

```

149 ADD COLUMN frequentFlyerID VARCHAR(20) NULL;
150
151 -- Make frequentFlyerID unique so it can be referenced by a foreign key
152 • ALTER TABLE Customer
153 ADD UNIQUE (frequentFlyerID);
154
155 -- Create the new Cost table to store itemised charges and optional frequent-flyer information
156 • CREATE TABLE Cost (
157     leaseID INT NOT NULL,
158     customerID INT NULL,
159     charge VARCHAR(100) NOT NULL,
160     cost DECIMAL(10,2) NOT NULL,
161     description VARCHAR(255),
162     frequentFlyerID VARCHAR(20) NULL,
163     points INT,
164     PRIMARY KEY (leaseID, charge),
165     FOREIGN KEY (leaseID) REFERENCES Lease(leaseID),
166     FOREIGN KEY (customerID) REFERENCES Customer(customerID)
167 );
168 • Select * FROM Cost;
169
170
171

```

The output pane shows two log entries:

#	Time	Action	Message	Duration / Fetch
89	09:56:25	ALTER TABLE Customer ADD UNIQUE frequentFlyerID	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.079 sec
90	09:56:40	CREATE TABLE Cost (leaseID INT NOT NULL, customerID INT NULL, charge VARCHAR(100) NOT NULL, ...)	Error Code: 1050. Table 'cost' already exists	0.015 sec

1.

-- Alter the Customer table to add the new frequentFlyerID column

ALTER TABLE Customer

ADD COLUMN frequentFlyerID VARCHAR(20) NULL;

-- Make frequentFlyerID unique so it can be referenced by a foreign key

ALTER TABLE Customer

ADD UNIQUE (frequentFlyerID);

-- Create the new Cost table to store itemised charges and optional frequent-flyer information

CREATE TABLE Cost (

 leaseID INT NOT NULL,

 customerID INT NULL,

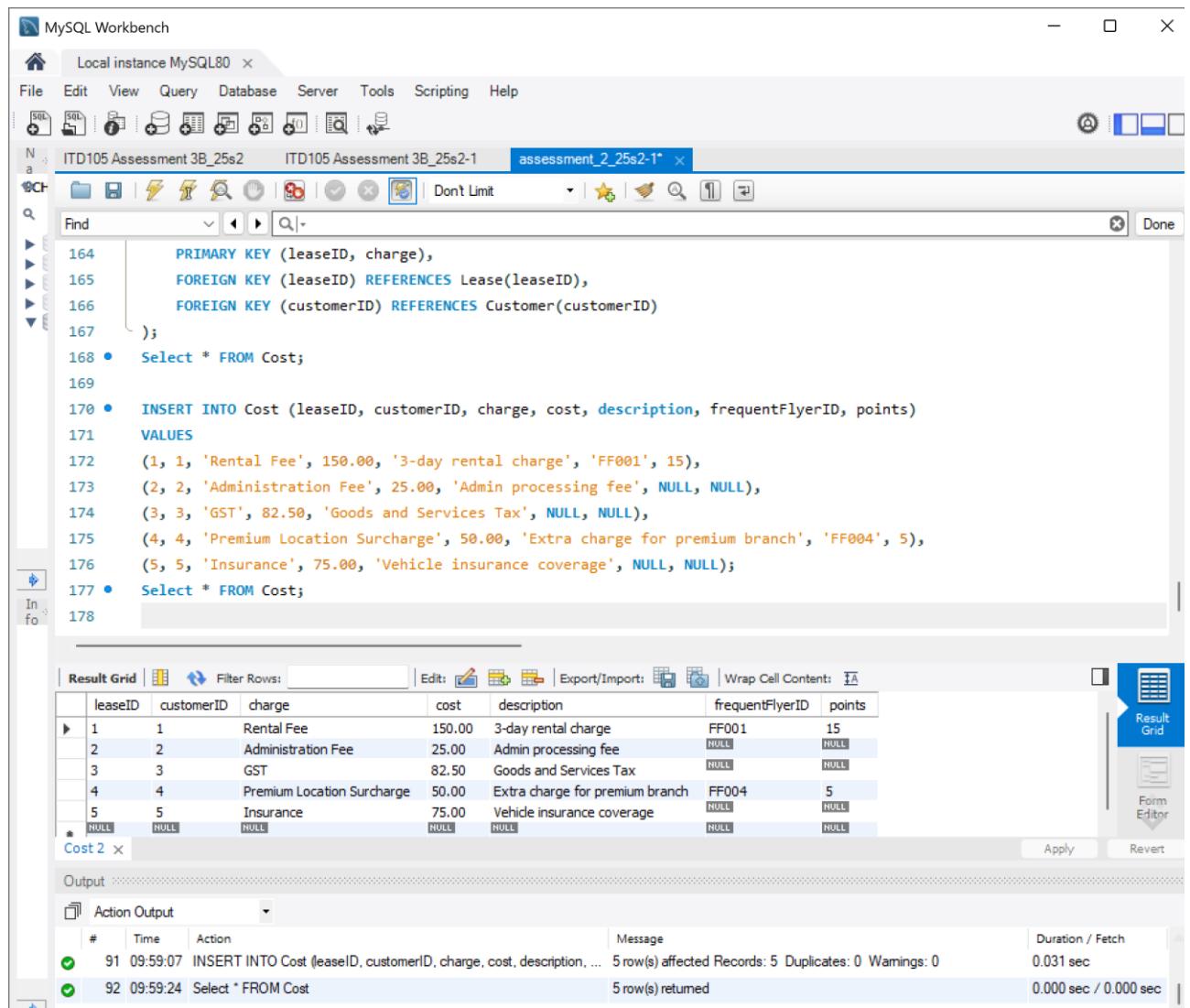
 charge VARCHAR(100) NOT NULL,

 cost DECIMAL(10,2) NOT NULL,

 description VARCHAR(255),

 frequentFlyerID VARCHAR(20) NULL,

points INT,
 PRIMARY KEY (leaseID, charge),
 FOREIGN KEY (leaseID) REFERENCES Lease(leaseID),
 FOREIGN KEY (customerID) REFERENCES Customer(customerID)
);



The screenshot shows the MySQL Workbench interface. The SQL tab contains the following code:

```

164     PRIMARY KEY (leaseID, charge),
165     FOREIGN KEY (leaseID) REFERENCES Lease(leaseID),
166     FOREIGN KEY (customerID) REFERENCES Customer(customerID)
167 );
168 • Select * FROM Cost;
169
170 • INSERT INTO Cost (leaseID, customerID, charge, cost, description, frequentFlyerID, points)
171     VALUES
172     (1, 1, 'Rental Fee', 150.00, '3-day rental charge', 'FF001', 15),
173     (2, 2, 'Administration Fee', 25.00, 'Admin processing fee', NULL, NULL),
174     (3, 3, 'GST', 82.50, 'Goods and Services Tax', NULL, NULL),
175     (4, 4, 'Premium Location Surcharge', 50.00, 'Extra charge for premium branch', 'FF004', 5),
176     (5, 5, 'Insurance', 75.00, 'Vehicle insurance coverage', NULL, NULL);
177 • Select * FROM Cost;
178

```

The Result Grid shows the following data:

leaseID	customerID	charge	cost	description	frequentFlyerID	points
1	1	Rental Fee	150.00	3-day rental charge	FF001	15
2	2	Administration Fee	25.00	Admin processing fee	NULL	NULL
3	3	GST	82.50	Goods and Services Tax	NULL	NULL
4	4	Premium Location Surcharge	50.00	Extra charge for premium branch	FF004	5
5	5	Insurance	75.00	Vehicle insurance coverage	NULL	NULL

The Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
91	09:59:07	INSERT INTO Cost (leaseID, customerID, charge, cost, description, frequentFlyerID, points)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
92	09:59:24	Select * FROM Cost	5 row(s) returned	0.000 sec / 0.000 sec

2.

INSERT INTO Cost (leaseID, customerID, charge, cost, description, frequentFlyerID, points)
 VALUES
 (1, 1, 'Rental Fee', 150.00, '3-day rental charge', 'FF001', 15),
 (2, 2, 'Administration Fee', 25.00, 'Admin processing fee', NULL, NULL),
 (3, 3, 'GST', 82.50, 'Goods and Services Tax', NULL, NULL),
 (4, 4, 'Premium Location Surcharge', 50.00, 'Extra charge for premium branch', 'FF004', 5),
 (5, 5, 'Insurance', 75.00, 'Vehicle insurance coverage', NULL, NULL);

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

SQL DDL DML Scripts Reports Find SQL Editor

ITD105 Assessment 3B_25s2 ITD105 Assessment 3B_25s2-1 assessment_2_25s2-1*

Find Done

```

165     FOREIGN KEY (leaseID) REFERENCES Lease(leaseID),
166     FOREIGN KEY (customerID) REFERENCES Customer(customerID)
167 );
168
169 •   Select * FROM Cost;
170
171 •   INSERT INTO Cost (leaseID, customerID, charge, cost, description, frequentFlyerID, points)
172     VALUES
173     (1, 1, 'Rental Fee', 150.00, '3-day rental charge', 'FF001', 15),
174     (2, 2, 'Administration Fee', 25.00, 'Admin processing fee', NULL, NULL),
175     (3, 3, 'GST', 82.50, 'Goods and Services Tax', NULL, NULL),
176     (4, 4, 'Premium Location Surcharge', 50.00, 'Extra charge for premium branch', 'FF004', 5),
177     (5, 5, 'Insurance', 75.00, 'Vehicle insurance coverage', NULL, NULL);
178
179 -- Primary Key violation example
180 •   INSERT INTO Branch (branchID, branchName, branchAddress, premiumLocationSurcharge)
181     VALUES ('B01010', 'Duplicate Branch', '123 Fake Street', FALSE);
182
183 -- foreign key violation
184 •   INSERT INTO Lease (vehicleID, pickupAt, rentalDate, returnedAt, returnDate, customerID, distanceKM)
185     VALUES ('XXXXXX', 'B01010', '2024-12-01', 'B01010', '2024-12-03', 1, 100);
186
187

```

Output

Action Output	#	Time	Action	Message	Duration / Fetch
	95	10:01:46	INSERT INTO Lease (vehicleID, pickupAt, rentalDate, returnedAt, r...	Error Code: 1452. Cannot add or update a child row: a foreign key c...	0.047 sec
	96	10:01:53	INSERT INTO Branch (branchID, branchName, branchAddress, pre...	Error Code: 1062. Duplicate entry 'B01010' for key 'branch.PRIMARY'	0.000 sec

3.

```
-- Primary Key violation example
INSERT INTO Branch (branchID, branchName, branchAddress,
premiumLocationSurcharge)
VALUES ('B01010', 'Duplicate Branch', '123 Fake Street', FALSE);

-- foreign key violation
INSERT INTO Lease (vehicleID, pickupAt, rentalDate, returnedAt, returnDate,
customerID, distanceKM)
VALUES ('XXXXXX', 'B01010', '2024-12-01', 'B01010', '2024-12-03', 1, 100);
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

ITD105 Assessment 3B_25s2 ITD105 Assessment 3B_25s2-1 assessment_2_25s2-1*

Find Done

```
184 • INSERT INTO Lease (vehicleID, pickupAt, rentalDate, returnedAt, returnDate, customerID, distanceKM)
185     VALUES ('XXXXXX', 'B01010', '2024-12-01', 'B01010', '2024-12-03', 1, 100);
186
187 • SET SQL_SAFE_UPDATES = 0;
188
189 • UPDATE Vehicle
190     SET luggageCapacity = 'Small'
191     WHERE model = 'Kia Stonic';
192
193 • SET SQL_SAFE_UPDATES = 1;
194     -- have to run safe and not safe mode when executing the above lines
195
196 • SELECT vehicleID, model, luggageCapacity
197     FROM Vehicle
198     WHERE model = 'Kia Stonic';
```

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

vehicleID	model	luggageCapacity
V01010	Kia Stonic	Small
V01110	Kia Stonic	Small
V01453	Kia Stonic	Small
NULL	NULL	NULL

Vehicle 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
102	10:05:17	UPDATE Vehicle SET luggageCapacity = 'Small' WHERE model = '...	1 row(s) affected Rows matched: 3 Changed: 1 Warnings: 0	0.047 sec
103	10:06:40	SELECT vehicleID, model, luggageCapacity FROM Vehicle WHERE...	3 row(s) returned	0.000 sec / 0.000 sec

4.

SELECT vehicleID, model, luggageCapacity

FROM Vehicle

WHERE model = 'Kia Stonic';

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL

ITD105 Assessment 3B_25s2 ITD105 Assessment 3B_25s2-1 assessment_2_25s2-1*

Find Done

```

194    -- have to run safe and not safe mode when executing the above lines
195    -- verify proof
196 •  SELECT vehicleID, model, luggageCapacity
197     FROM Vehicle
198     WHERE model = 'Kia Stonic';
199
200 •  SELECT
201      b.branchID,
202      b.branchName,
203      COUNT(l.leaseID) AS numPickups
204  FROM Branch b
205  LEFT JOIN Lease l ON b.branchID = l.pickupAt
206  GROUP BY b.branchID, b.branchName
207  ORDER BY numPickups DESC;
208

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result 4 x | Read Only

branchID	branchName	numPickups
B01010	Mooloolaba	6
B17454	Adelaide Airport	4
B17412	Kingston	4
B14236	Legana	4
B01120	Cairns City	3
R19161	Fatm	3

Action Output

#	Time	Action	Message	Duration / Fetch
103	10:06:40	SELECT vehicleID, model, luggageCapacity FROM Vehicle WHERE...	3 row(s) returned	0.000 sec / 0.000 sec
104	10:07:47	SELECT b.branchID, b.branchName, COUNT(l.leaseID) A...	13 row(s) returned	0.000 sec / 0.000 sec

5.

SELECT

b.branchID,

b.branchName,

COUNT(l.leaseID) AS numPickups #counts how many times a vehicle was picked up from each branch.

FROM Branch b

LEFT JOIN Lease l ON b.branchID = l.pickupAt #links each branch to its leases (pickups). if theres a match lease row fills if not NULL. Pickup is matching key

GROUP BY b.branchID, b.branchName #groups results by branch.

ORDER BY numPickups DESC; #shows the branches with the most pickups first.

6.

The screenshot shows the MySQL Workbench interface. The query editor contains a script named 'assessment_2_25a2-1' with the following SQL code:

```
197 FROM Vehicle
198 WHERE model = 'Kia Stonic';
199
200 • SELECT
201     b.branchID,
202     b.branchName,
203     COUNT(l.leaseID) AS numPickups #counts how many times a vehicle was picked up from each branch.
204     FROM Branch b
205     LEFT JOIN Lease l ON b.branchID = l.pickupAt #links each branch to its leases (pickups). if theres a match lease row fills if not NULL. Pickup is matching key
206     GROUP BY b.branchID, b.branchName #groups results by branch.
207     ORDER BY numPickups DESC; #shows the branches with the most pickups first.
208
209 -- I had to Update the lease to reference a valid branch to delete it because Branch is a parent table in a foreign key relationship
210 • UPDATE Lease
211     SET pickupAt = 'B10245' -- doesnt matter, any valid branch
212     WHERE pickupAt = 'TESTBR';
213
214 -- Delete the fake test branch
215 • DELETE FROM Branch
216     WHERE branchID = 'TESTBR';
217
218
219
```

The output pane shows the results of the executed statements:

Action	Time	Message	Duration / Fetch
UPDATE Lease	107 10:25:53	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
DELETE FROM Branch	108 10:25:53	1 row(s) affected	0.000 sec

-- I had to Update the lease to reference a valid branch to delete it because Branch is a parent table in a foreign key relationship

UPDATE Lease

SET pickupAt = 'B10245' -- doesnt matter, any valid branch

WHERE pickupAt = 'TESTBR';

-- Delete the fake test branch

DELETE FROM Branch

WHERE branchID = 'TESTBR';

7.

The screenshot shows the MySQL Workbench interface with a query editor window. The query is as follows:

```
216 WHERE branchID = 'TESTBR';
217
218 • SELECT DISTINCT v.vehicleID, v.numberPlate
219 FROM Vehicle v
220 JOIN Lease l ON v.vehicleID = l.vehicleID #link vehicles to their leases and ensure only vehicles with one lease appears.
221 WHERE l.returnDate BETWEEN '2024-11-01' AND '2024-11-30' #select only November returns.
222 AND l.distanceKM > 3000 #only over 3000km
223 AND v.vehicleID NOT IN (
224     SELECT vehicleID
225     FROM Lease
226     WHERE returnDate IS NULL #check if vehicles are on lease and returns only the ones that aren't from NOT IN.
227 )
```

The result grid shows two rows of data:

vehicleID	numberPlate
V04514	WER564
V01110	AWQ323

The results pane shows the following log entries:

#	Action	Time	Message	Duration / Fetch
108	DELETE FROM Branch	10:25:53	branchID = 'TESTBR'	1 row(s) affected 0.000 sec
109	SELECT DISTINCT v.vehicleID, v.numberPlate	10:41:36	FROM Vehicle v JOIN Lease l ON v.vehicleID = l.vehicleID #link vehi... AND l.distanceKM > 3000 #only over 3000km AND v.vehicleID NOT IN (2 row(s) returned 0.000 sec / 0.000 sec

Query Completed

SELECT DISTINCT v.vehicleID, v.numberPlate

FROM Vehicle v

JOIN Lease l ON v.vehicleID = l.vehicleID #link vehicles to their leases and ensure only vehicles with one lease appears

WHERE l.returnDate BETWEEN '2024-11-01' AND '2024-11-30' #select only November returns.

AND l.distanceKM > 3000 #only over 3000km

AND v.vehicleID NOT IN (

 SELECT vehicleID

 FROM Lease

 WHERE returnDate IS NULL #check if vehicles are on lease and returns only the ones that aren't from NOT IN.

