Solutions

Chapter 20: Project: Using SQL to query a database

Knowledge probe: Querying the *Movie_data* file, page 125

1
```
SELECT Movie_data.Movie_title, Movie_data.Director_name,
Movie_data.Budget
FROM Movie_data
WHERE Movie_data.Budget > 300000000
ORDER BY Movie_data.Budget DESC;
```

2
```
SELECT Movie_data.Movie_title, Movie_data.Director_name,
Movie_data.Budget, Movie_data.Gross
FROM Movie_data
WHERE Movie_data.Gross > 400000000
ORDER BY Movie_data.Gross DESC;
```

3
```
SELECT Movie_data.Movie_title, Movie_data.Director_name,
Movie_data.Actor_1_name, Movie_data.Budget,
Movie_data.Gross
FROM Movie_data
WHERE Movie_data.Gross<Budget
ORDER BY Movie_data.Budget DESC;
```

4 This is done by adding fields at the top of the query after SELECT, e.g.
```
SELECT Movie_data.Actor_1_name, Movie_data.Title_year
```

5 Students' responses will vary.

Knowledge probe: Joined file queries, page 127

1
```
SELECT Movie_data.Movie_ID, Movie_data.Movie_title,
Customer_data.First_name, Customer_data.Surname
FROM (View_data INNER JOIN Customer_data ON
View_data.Customer_ID = Customer_data.Customer_ID) INNER
JOIN Movie_data ON View_data.Movie_ID = Movie_data.Movie_ID
WHERE Movie_data.Movie_ID = "MOV-0009";
```

2
```
SELECT Movie_data.Movie_ID, Movie_data.Actor_1_name,
Movie_data.Movie_title, Customer_data.First_name,
Customer_data.Surname
FROM (View_data INNER JOIN Customer_data ON
View_data.Customer_ID = Customer_data.Customer_ID) INNER
JOIN Movie_data ON View_data.Movie_ID = Movie_data.Movie_ID
WHERE Movie_data.Actor_1_name = "Johnny Depp";
```

3
```
SELECT Movie_data.Movie_ID, Movie_data.Director_name,
Movie_data.Movie_title, Customer_data.First_name,
Customer_data.Surname
FROM (View_data INNER JOIN Customer_data ON
View_data.Customer_ID = Customer_data.Customer_ID) INNER
JOIN Movie_data ON View_data.Movie_ID = Movie_data.Movie_ID
WHERE Movie_data.Director_name = "Tim Burton";
```

4
```
SELECT Movie_data.Movie_ID, Movie_data.Director_name,
Movie_data.Movie_title, Customer_data.First_name,
Customer_data.Surname
FROM (View_data INNER JOIN Customer_data ON
View_data.Customer_ID = Customer_data.Customer_ID) INNER
JOIN Movie_data ON View_data.Movie_ID = Movie_data.Movie_ID
WHERE Movie_data.movie_title LIKE  "Har*";
```

5

a
```
SELECT Customer_data.Customer_ID,
Customer_data.First_name, Customer_data.Surname,
Customer_data.DOB,
DateDiff("yyyy",Customer_data.DOB,Now()) AS Age
FROM Customer_data;
```
This query should be saved as Customer_Age.

b

- Mostly *Spiderman* movies. There seems to be a pattern here.
- The two oldest customers are both 57. There does not seem to be an obvious pattern to their viewing. The student would need to investigate further by selecting more fields in the view. These two customers appear to enjoy watching romance movies.

  The following code can be used to return the age order for old-to-young or vice versa, by changing DESC to ASC.
```
SELECT Customer_data.First_name, Customer_data.Surname,
Customer_Age.Age, Movie_data.Movie_title
FROM ((View_data INNER JOIN Customer_data ON
View_data.Customer_ID = Customer_data.Customer_ID) INNER
JOIN Movie_data ON View_data.Movie_ID =
Movie_data.Movie_ID) INNER JOIN Customer_Age ON
Customer_data.Customer_ID = Customer_Age.Customer_ID
ORDER BY Customer_Age.Age DESC;
```
- This is an action/adventure movie, which is highly suited to a male teenage audience.
- This is an older movie (1969), which would suit a more mature audience. Nostalgic viewing.

6 Survey fields could include: favourite movies, favourite genres, favourite actors, recently watched movies, etc.

7 Discussion could include: data breaches, privacy breaches, data on-selling, privacy policies, and federal and state privacy laws.

8 Students' responses will vary.