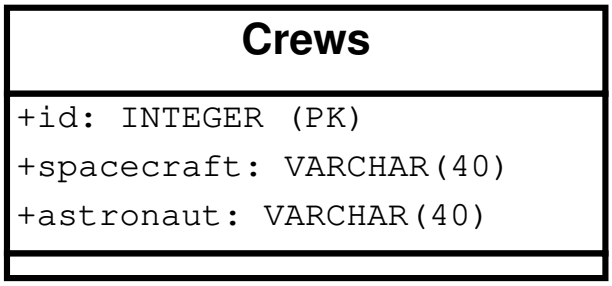
Graded Exercise 1

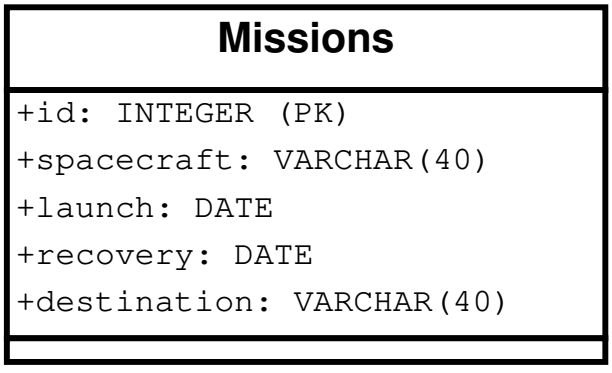
CSC230 Database Technologies for Analytics

01 November 2019

Schema




```
CREATE TABLE crews (
  `index` INTEGER PRIMARY KEY,
  `spacecraft` VARCHAR(40) NOT NULL,
  `astronaut` VARCHAR(40) NOT NULL
);
```



```
CREATE TABLE `missions` (
    `index` INTEGER PRIMARY KEY,
    `spacecraft` VARCHAR(40) NOT NULL,
    `launch` DATE NOT NULL,
    `recovery` DATE NOT NULL,
    `destination` VARCHAR(40) NOT NULL
);
```

Questions

Explain what each of these SQL SELECT statements accomplishes (for example, "produces a table that contains the names of all astronauts and their spacecraft"). Identify the meaning of symbols (for example, * and %). Identify the functions (for example, COUNT, SUBSTR, TO_DAYS, YEAR). Although you need not define every reserved word of the SQL language, some explanations will require that you show an understanding of the roles of various features of the language (for example, AS, GROUP BY, ORDER BY).

1. `SELECT * FROM crews;`

Produce a table that contains 3 columns:

(a) the number of the record

(b) the name of a spacecraft

(c) the name of an astronaut

This table will contain all of the records in the crews table.

2. `SELECT * FROM missions;`

Produce a table that contains 5 columns:

(a) the number of the record

(b) the name of a spacecraft

(c) the date of the spacecraft's launch

(d) the date of the spacecraft's return to earth

(e) the spacecraft's destination

This table will contain all of the records in the missions table.

3.
```
SELECT * FROM crews
    WHERE spacecraft LIKE 'Apollo%';
```

Produce a table that contains 3 columns:

(a) the number of the record

(b) the name of a spacecraft

(c) the name of a restaurant

This table will contain records in the crews table in which the name of the spacecraft begins with 'Apollo.'

4.
```
SELECT * FROM crews
    WHERE CAST(SUBSTR(spacecraft, 7) AS UNSIGNED) >= 11;
```

Produce a table that contains 3 columns:

(a) the number of the record

(b) the name of a spacecraft

(c) the name of an astronaut

This table will contain records in the crews table in which the name of the spacecraft ends with a number that is greater than or equal to 11. The query searches for a substring in the name that begins at position 7 and continues to the end of the name. It then converts (or attempts to convert) that substring to a non-negative whole number.

5.
```
SELECT destination, COUNT(destination)
    FROM missions
    GROUP BY destination;
```

Produce a table that contains 2 columns:

(a) the destination of a spaceflight

(b) the number of spaceflights with that destination

COUNT is an aggregate function.

GROUP BY is needed to aggregate the data that is to be counted.

The query draws this data from the missions table.

6.
```
SELECT spacecraft,
        (TO_DAYS(recovery) - TO_DAYS(launch)) AS 'duration'
    FROM missions;
```

Produce a table that contains 2 columns:

(a) the name of a mission to space

(b) the number of whole days that spacecraft spent in space

TO_DAYS is a date and time function.

It converts a date to a number of days since the first day of the (imaginary) year 0. This conversion allows subtraction of the date on which a spacecraft returned to earth from the date on which it left earth.

The query gets the data from the missions table.

7.
```
SELECT * FROM crews
    WHERE astronaut IN ('Aldrin', 'Armstrong', 'Collins')
    ORDER BY astronaut ASC, spacecraft DESC;
```

Produce a table that contains 3 columns:

(a) the number of the record

(b) the name of a spacecraft

(c) the name of an astronaut

The table includes only those records in which the name of the astronaut is Aldrin, Armstrong, or Collins. These astronauts flew together on the first mission that put astronauts on the moon. This table shows all of the spaceflights on which the three astronauts flew, including the Apollo 11 moon landing mission.

The query orders the results firstly in ascending alphabetic order by the name of the astronaut and then secondly in descending alphabetic order by the name of the spacecraft.

The query draws the data from the crews table.

8.
```
SELECT c.astronaut, m.destination
    FROM crews AS c INNER JOIN missions AS m
    USING (spacecraft) ORDER BY astronaut, destination;
```

Produce a table with 2 columns:

(a) the name of an astronaut

(b) the name of destination of the destination of one of that astronaut's journeys into space

The query orders the results in ascending alphabetic order firstly by the name of the astronaut and then by the name of the destination. The query draws data from both the crews table and the missions table.

9.
```
SELECT astronaut, COUNT(astronaut) AS 'flights'
    FROM crews
    GROUP BY astronaut
    ORDER BY 'flights' DESC, astronaut ASC;
```

Produce a table that contains 2 columns:

(a) the name of an astronaut

(b) the number of times that that astonaut flew into space

The query orders the records firstly in descending numerical order by number of spaceflights and secondly in ascending alphabetic order by the names of the astronauts. The query draws data from the crews table.

COUNT is an aggregate function.

GROUP BY is needed to aggregate (place in adjacent records) the records for each astronaut.

10.
```
SELECT a.astronaut, b.astronaut
    FROM crews AS a INNER JOIN crews AS b
    USING (spacecraft)
    WHERE a.astronaut <> b.astronaut
    ORDER BY a.astronaut, b.astronaut;
```

Produce a table that contains 2 columns:

(a) the name of an astronaut

(b) the name of another astronaut

The query draws data from 2 copies of the crews table. For each record in the first copy, it pairs a record in the second copy that lists the same spacecraft. In this way, it identifies for each astronaut A all of the other astronauts who flew with A.

The WHERE clause selects rows from the table. In this case, it selects all rows except those that contain the name of the same astronaut in in the first and second copies of the crews table.

The query orders records firstly in ascending alphabetic order of the names of the astronauts in the first column, and secondly in ascending alphabetic order of the names of the astronauts in the second column.

11.
```
SELECT a.astronaut, COUNT(a.astronaut) AS 'crewmates'
    FROM crews AS a INNER JOIN crews AS b USING (spacecraft)
    WHERE a.astronaut <> b.astronaut
    GROUP BY a.astronaut
    ORDER BY 'crewmates' DESC, a.astronaut ASC;
```

Produce a table that contains 2 columns:

(a) the name of an astronaut

(b) the number of other astronauts with whom that astronaut flew

The query draws data from 2 copies of the crews table. The query orders records firstly in descending numerical order by the number of crewmates and secondly in ascending alphabetical order by the names of the astronauts.

12.
```
SELECT YEAR(launch) AS 'year', COUNT(launch) AS 'flights'
    FROM missions
    GROUP BY YEAR(launch)
    ORDER BY 'year';
```

Produce a table that contains 2 columns:

(a) a year in which Americans flew into space

(b) the number of American manned spaceflights in that year

The query draws data from the missions table.

YEAR is a data and time function. It extracts the year from a date.

COUNT is an aggregate function. The query orders records in ascending numerical order by year.