Name: Class:

Task 1

The following shows a database table named Customers. This is used in an activity centre. All activities need to cost less that £100 and activities must be 2 hours or less.

Table name: **Activities**

Table

Description automatically generated

(a) Match each of the fields below with the validation rule that would be most appropriate.

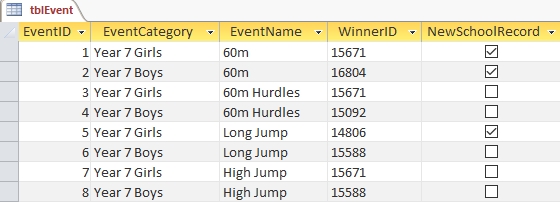
|  |  |  |
| --- | --- | --- |
| **Field** |  | **Validation rule** |
| ActivityName |  | <100.00 |
| NumberOfPeople | [NumberOfPeople]<=[MaxNumOfPeople] |
| Cost | <=120 |
| MaxNumOfPeople | Length is 255 characters or fewer |
| LengthInMins | <30 |

(b) ActivityID is the primary key for the table.

Explain what the term primary key means.

Task 2

Two tables, **tblEvent** and **tblPupil**, are shown below.



Table

Description automatically generated

1. Write SQL statements to:

(a) select the “Firstname”, “Surname” and “DateOfBirth” columns from tblPupil.

(b) select the “EventCategory” and “EventName” columns from tblEvent for which new school records were set.

(c) select all columns from tblPupil for pupils in Darwin, Faraday or Bell.

(d) select all columns from tblEvent for which for which the Event category is Year 7 girls and the event name is either 60m or 60m Hurdles.

2. List the data that could be displayed after executing the following SQL queries on the database.

(a) SELECT FirstName, Surname, House

FROM tblPupil

WHERE House = 'Darwin'

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

(b) SELECT EventCategory, EventName

FROM tblEvent

WHERE EventName = 'Long Jump' OR EventName = 'High Jump'

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Task 3

(a) Write an SQL statement to sort **tblEvents** in Task 2 into ascending order by EventCategory, displaying only the fields EventCategory, EventName and WinnerID.

(b) Write an SQL statement to sort **tblPupil** in Task 2 into descending order by Surname, displaying only the fields Firstname and Surname

**Task 4**

The following table, named Passengers, shows the data for passengers who have checked in for a flight.

Table

Description automatically generated

(a) Write an SQL statement to show the total value of all tickets for the flight.

(b) Write an SQL statement to show the total check in luggage weight.

(c) Write an SQL statement to count the number of passengers on the flight.