Higher

Computing Science

Database Design and Development

Implementation - Wildcards

Name:_____

Contents

N5 Query Revision

Select Queries

```
SELECT Assessment.AssessmentType, Assessment.Title, Assessment.Marks
FROM Assessment
WHERE Assessment.AssessmentType = “Multiple Choice” AND Assessment.Marks <=100
ORDER BY Assessment.AssessmentType DESC, Assessment.Title ASC;
```

Select Queries (Equi-Join)

```
SELECT Student.Forename, Student.Surname, Result.Mark
FROM Student,Result
WHERE Student.StudentNo = Result.StudentNo AND  Result.Mark > 60
```

Update Queries

```
UPDATE Tutor Group
SET Room = 'D15'
WHERE TGCode = ‘TG5’;
```

Insert Queries

```
INSERT INTO Result (StudentNo, AssessmentCode, Mark, AssessmentDate)
VALUES ('101237','X216','99', '2016-10-23');
```

Delete Queries

```
DELETE FROM Result WHERE Result.StudentNo = "101237"
```

Higher Queries

You must be able to implement UPDATE, SELECT, DELETE and INSERT queries making use of:

- Wildcards
- Aggregate Functions (MIN, MAX, AVG, SUM, COUNT)
- Computed Values, Alias
- GROUP BY
- ORDER BY
- WHERE

Wildcards

A wildcard character is used to replace one or more characters in a string.

Wildcards are useful in situations when incomplete information is available and it would be impossible to write a WHERE clause using one of the existing logical operators =, <, >, ≤ or ≥.

Wildcard characters are used with the SQL LIKE operator. The LIKE operator is used in a WHERE clause to perform search operations.

Two different wildcards can be used:

% (the percent symbol) is used to represent zero, one or multiple characters

_ (the underscore symbol) is used to represent a single character

Note: MS Access uses an asterisk (*) rather than % and a question mark (?) rather than __

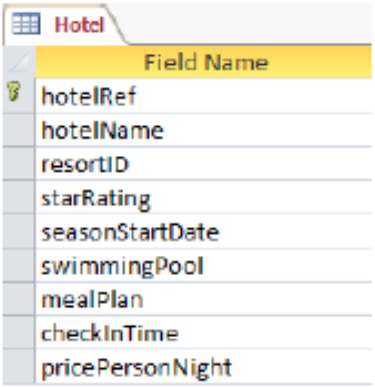
The following are some examples of LIKE used with the wildcards:

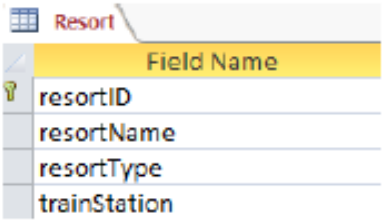
Example	Purpose
`WHERE surname LIKE 'Thom%'`	Used to find any values in the surname field that start with "Thom"
`WHERE surname LIKE '%son'`	Used to find any values in the surname field that end with "son"
`WHERE surname LIKE '%is%'`	Used to find any values that have "is" anywhere in the surname field
`WHERE surname LIKE '_h%'`	Used to find any values in the surname field that have "h" as the second character
`WHERE surname LIKE 'm % %'`	Used to find any values in the surname field that start with "m" and have at least 3 characters
`WHERE surname LIKE 'a%z'`	Used to find any values in the surname field that that start with "a" and end with "z"

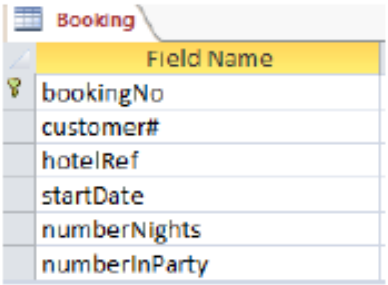
Example:

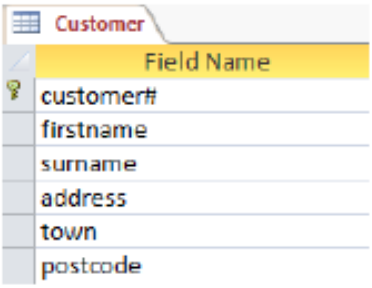
A travel agency uses a relational database to enable their employees to view details of hotels in Scottish holiday resorts and make bookings for customers. The details are stored in four separate tables called Hotel, Resort, Booking and Customer.

The structure of these tables is shown below:









Example 1: used to search the database to display the name, swimming pool details, resort and resort type of any hotel in a coastal resort that starts with the letter 'A'.

```
SELECT hotelName, swimmingPool, resortName, resortType

FROM Hotel, Resort

WHERE Hotel.resortID = Resort.resortID AND resortName LIKE 'A*' AND resortType = 'coastal';
```

Example 2: used to display the customer's full name, booking number, start date, hotel name and resort name of all customers who has an 'h' as the second letter of their surname. These details should be listed in alphabetical order of surname; customers with the same surname should be listed so that the customer with the earliest holiday should be listed first.

```
SELECT firstname, surname, bookingNo, hotelName, resortName, startDate

FROM Customer, Booking, Hotel, Resort

WHERE Customer.[customer#]=Booking.[customer#] AND
Booking.hotelRef=Hotel.hotelRef AND Hotel.resortID=Resort.resortID AND surname
LIKE '?h*'


ORDER BY surname ASC, startDate ASC;
```

Worked Examples – Wildcards

Task 1a

Design a query to display all assessment data for any assessment type ending in the work ‘answer’

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentType like '*answer'
Grouping	
Sort Order	

```
SELECT *

FROM Assessment

WHERE Assessment.[AssessmentType] Like '*answer';
```

Task 1b

Design a query to display all assessment data for any assessment with a title starting with ‘data’ with the shortest assessment listed first

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	Title like 'data*'
Grouping	
Sort Order	duration ASC

```
SELECT *

FROM Assessment

WHERE Title like 'data*'

ORDER BY duration ASC;
```

Task 1c

Design a query to display all assessment data for any assessment where the writerr's name contains the letters 'men' in reverse alphabetical order by title

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentType like '*men*'
Grouping	
Sort Order	Title DESC

```
SELECT *

FROM Assessment

WHERE writer like '*men*'

ORDER BY title DESC;
```

Task 1d

Design a query to display all assessment data for any assessment where the assessment code has exactly 3 characters

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentCode like '???'
Grouping	
Sort Order	

```
SELECT *

FROM Assessment

WHERE assessmentcode like '???';
```

Task 1e

Design a query to display all assessment data for any assessment where the second word in the assessment type has exactly 6 characters. List these details in alphabetical order by assessment type; listing assessments with the same type in order of longest duration first.

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentType like '* ??????'
Grouping	
Sort Order	AssessmentType ASC, duration DESC

```
SELECT *

FROM Assessment

WHERE assessmentType like '* ??????'

ORDER BY assessmentType ASC, duration DESC;
```

Task 1f

Design a query to display all assessment data for any assessment where the second character in the assessment code is a ‘2’.

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentType like '?2*'
Grouping	
Sort Order	

```
SELECT *

FROM Assessment

WHERE assessmentcode like '?2*';
```

Task 1g

Design a query to display all assessment data for any assessment with at least 4 characters where the second character in the assessment code is a '1'.

Field(s) and calculation(s)	AssessmentCode, Title, AssessmentType, duration, writer, passMark
Table(s) and query	Assessment
Search criteria	AssessmentType like '?1*?*?'
Grouping	
Sort Order	

```
SELECT *

FROM Assessment

WHERE assessmentcode like '?1*?*?';
```

Practise Questions - Wildcards

Task 1

Which of the following strings satisfies the condition `'T_'`? (There can be more than one answer)

`TW`

`TWITTER`

`TAIWAN`

`To`

Which of the following strings satisfies the condition `'A_B%'`? (There can be more than one answer)

`AKKB`

`AKBKKK`

`ABKK`

`ABBB`

Which of the following strings satisfies the condition `'Z%K_R'`? (There can be more than one answer)

`ZKRR`

`ZYKR`

`ZRKKJ`

`ZABCDEFKGR`

Write the conditions used to list `productName`s that -

start with the letter `'S'`.

end with the letters `'on'`.

contain the pattern `'ill'`.

contain the letters `'o'` and `'n'` (in that order) with exactly one letter between them.

contain the patterns `'aw'` and `'ing'` separated by exactly two letters.

start with the pattern `'St'` which is followed by any character which in turn is followed by the letter `'a'` and ends with the letter `'s'`.

Task 2
A table called Sales is shown below.

firstName	lastname	birthdate	joinDate	totalSales
Sophie	Lee	05/01/1960	05/04/2015	500
Richard	Brown	07/01/1975	05/04/2015	200
Jamal	Santo	08/10/1983	09/04/2015	350
Casey	Healy	20/09/1996	09/04/2015	80
Jill	Wilkes	20/11/1979	15/04/2015	210

Which of the following SQL statement is valid? (There can be more than one answer)

```
SELECT firstName, lastName
FROM Sales
WHERE firstName LIKE 'A%' lastName LIKE 'W%';

SELECT firstName, lastName
FROM Sales
WHERE firstName LIKE 'J%' AND lastName LIKE 'W%';

SELECT firstName, lastName
FROM Sales
firstName LIKE 'J%' AND lastName LIKE 'W%';

SELECT firstName, lastName
FROM Sales
WHERE firstName LIKE 'J%', lastName LIKE 'W%';
```

How many records will be returned by the following query? (Assuming the database is not case sensitive)

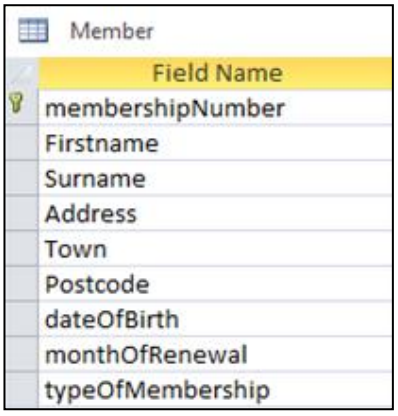
```
SELECT *
FROM Sales
WHERE lastName LIKE '%l_e%';
```

How many records will be returned by the following query? (Assuming the database is not case sensitive)

```
SELECT *
FROM Sales
WHERE firstName LIKE '%a%'
OR lastName LIKE '%e%';
```

Practical Task - Wildcards

Open the file called Members Database. This database has one table called Member.



Member

Field Name
membershipNumber
Firstname
Surname
Address
Town
Postcode
dateOfBirth
monthOfRenewal
typeOfMembership

Create SQL queries to display the required details.

1. List the full name, postcode and the renewal month of all members with a surname that begins with the letter 'D'.

2. List the membership number, surname and full postal address of all members who have a surname that contains the pattern 'oo'. List these members in alphabetical order of surname.

3. List the first name and home town of members with a first name that contains the letter 'o' and who live in a town that starts with the letter 'B' and ends with the letter 'n'.

4. List the full name of any members who have a surname that has exactly 4 letters. starting with the letter 'L'.

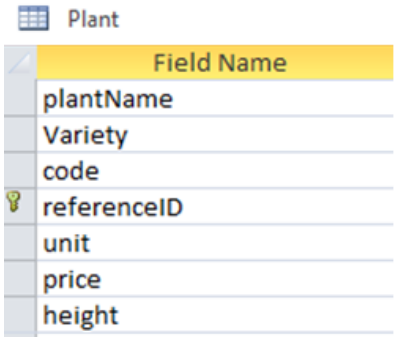
5. List the membership number, town and postcode of all members with a postcode that contains the letter 'a' and the digit '2' separated by exactly 1 character.

6. List the full name and type of membership of all members who have a surname that contains the letters 'I' and 'e' separated by exactly two characters.

7. List the membership number, date of birth and the type of membership for all adult members who were born in the month of October.

Task 2

Open the file called Plants Database. This database has one table called Plant.



Plant

Field Name
plantName
Variety
code
referenceID
unit
price
height

Create SQL queries to display the required details.

1. List the name and variety of any plant that has the letter 'x' somewhere in the name of its variety. These details should be listed in alphabetical order of plant name.

2. List the category, name, plant code and price of any plant with a plant code that has exactly 2 characters.

3. List the name of any plant, together with its plant code and height, with a code that contains the letter 'P' and a plant name that contains the letters 'a' and 't' separated by exactly one character.

4. List the name, reference ID and price of all plants with an 'a' near the start of its name which also ends with the letter 'a'. The dearest plant should be listed first; plants with the same price should be listed in alphabetical order of referenceID.

5. List the plant code, referenceID and category of any plant that has a '3' in the middle of its referenceID (each referenceID has 3 characters) and the letter 'r' anywhere in its plant code. These plants should be listed in alphabetical order of category.

6. List the plant name, unit size and price of any plant with the letters 'a' and 'n' (separated by exactly 2 other letters) in its name. Arrange these plant details so that the largest unit size is listed first; plants with the same unit size should be arranged so that the cheapest plant is listed first.

7. List the referenceID, plant name, variety and height of any plant with the letter 'e' as the second letter of its plant name. Only plants the belong to a variety that starts with the letter 'C' and ends with the letter 'e' should be listed.