Databases II:
Microsoft Access

CS111, 2016
Recapitulation

A database is a collection of data that is **systematically organized**, so as to allow efficient **addition, modification, removal** and **retrieval**.

A relational database is a collection of **tables**, where each row of the table is a **record** and each column is a **field**.

Databases use **foreign keys** and **primary keys** to establish **relationships** between records on different tables.

A database has **referential integrity** when all of the values in all foreign key fields point to the primary key of an existing record in the appropriate table (or are null).
Today

Microsoft Access

- Creating a database and adding data
- Querying that database, i.e. retrieving specific information from the database
- If time allows, a brief look at report generation
Microsoft: Access

Creating a Database
Zoo Database

I am going to build a database that a Zoo will use to keep track of the offspring of an endangered species. It is a deliberately simple database.

Looking at these tables (and using some common sense about mothers and offspring) you should be able to figure out

- the relationship between these tables
- what the primary and foreign keys are.

<table>
<thead>
<tr>
<th>TABLE: Mothers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE: Offspring</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>
Creating the database

1. Create the database

2. Create the tables, specifying details about each of the fields in each table

3. “Populate the database” i.e. add data to the database
Creating a database

Templates

Creating a new database
Creating a table

New table button
Creating a table

- **Design view:** create/view the fields in the table
- **Datasheet view:** create/view data in the table
Design view

The data type determines the kind of values that users can store in the field. Press F1 for help on data types.
Reassigning the primary key

Right clicking on the field allows you to assign the primary key to something different than the default.
Datasheet view

Allows us to enter data into our table

Some checks are done to ensure that we enter the correct type of data in each field (eg. no text in a number field)
Creating a relationship

Relationships view allows us to create relationships between fields in different tables.

Database Tools tab → Relationships button
Creating a relationship
Creating a relationship
Creating relationships
Inserting data

- Can we insert this record in the Enrolments table?
Inserting data

- This won’t work; StudentID’s value (‘5’) doesn’t exist in the primary key ID
Retrieving information from a DB

Queries
Queries

Queries allow you to retrieve certain **fields** from certain **records** from your database.

There are two kinds of queries in Access:

**Query by example (QBE)**
Visual way of designing queries

**SQL (Structured Query Language):**
Uses commands to retrieve data from databases.
Developed by IBM in the late 1970’s.

* Access actually converts QBE queries into SQL before running them.
Query By Example (QBE)

We will “ask the database” these questions...

A. What are the birth-weights of all of the offspring?
B. What are the nicknames of all of the offspring of Beauty?
C. In alphabetical order, what are the nicknames of all of the offspring of Beauty?
D. In alphabetical order, what are the nicknames and birth-weights of all of Beauty’s offspring?
E. In alphabetical order, what are the nicknames of and birth-weights of all of Beauty’s offspring … that have a birth-weight of over 51 kilos?
F. What is the average birthweight of all of Bea’s offspring?
QBE queries

1. Select Query Design from the Create Menu

2. Select tables to use in query
Choosing fields

Adding criteria to the field

QBE queries

QBE grid
QBE queries

‘Run’ button

Query results

<table>
<thead>
<tr>
<th>Country</th>
<th>Reserve</th>
<th>Land Area</th>
<th>Water Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Zealand</td>
<td>Arthurs Pass National Park</td>
<td>268,670</td>
<td>10</td>
</tr>
<tr>
<td>New Zealand</td>
<td>Mount Aspiring National Park</td>
<td>268,670</td>
<td>10</td>
</tr>
<tr>
<td>New Zealand</td>
<td>South Taupo Wetland</td>
<td>268,670</td>
<td>10</td>
</tr>
</tbody>
</table>
QBE queries - sorting

- Results from QBE queries can be sorted in ascending and descending order
QBE queries

- A Totals QBE query allows us to group data using functions such as Min, Max, Avg, Sum etc.
QBE queries
QBE Exercise

Complete this QBE grid so that it will return the first names, surname and grade (in that order) of all students who have received an A+. Sort the results by surname in alphabetical order.
QBE Exercise Answer

Complete this QBE grid so that it will return the first names, surname and grade (in that order) of all students who have received an A+. Sort the results by surname in alphabetical order.
SQL introduction

Structured Query Language (SQL) was developed by IBM in the 1970s and is commonly used today.

It uses text commands to perform operations on databases, such as inserting and removing records and running queries.
QBE queries

```sql
WHERE (((Country.Country) = "New Zealand") AND (((Country.[Land Area] > 100000)));
SQL queries

Four clauses that can be part of a simple SQL query:

- SELECT
- FROM
- WHERE
- ORDER BY

We will now together develop an SQL query that returns the first names, surname, and grade (in that order) of all students who have received an A+, with the results sorted by surname in alphabetical order.
SQL queries - SELECT

Selects fields from the tables that we want to display in our results table

Syntax:

SELECT [comma separated list]

SELECT [First Names], Surname, Grade

Note the square brackets around ‘First Names’ needed because of the space

“We will now together develop an SQL query that returns the first names, surname, and grade (in that order) of all students who have received an A+, with the results sorted by surname in alphabetical order.”
SQL queries - FROM

Specifies the table which holds the field(s) listed in the SELECT clause

Syntax
FROM [comma separated list]

SELECT [First Names], Surname, Grade
FROM Students

“We will now together develop an SQL query that returns the first names, surname, and grade (in that order) of all students who have received an A+, with the results sorted by surname in alphabetical order.”
SQL queries - WHERE

Optional; used to provide criteria that limit the records displayed in the results table

Syntax
WHERE [criteria], [criteria], ...

There are a range of criteria we can use →

Comparisons (=, >, <, <=, >=, <>)
e.g., WHERE [Land Area] < 50000

BETWEEN ... AND ...
e.g., WHERE Price BETWEEN 10 AND 20

LIKE (some pattern)
e.g., WHERE [City] LIKE ‘San *’

AND, NOT, OR (combined with any of above)
e.g., WHERE Country = ‘New Zealand’ AND City = ‘Auckland’

IS NULL, IS NOT NULL
e.g., WHERE [Postal Code] IS NOT NULL
SQL queries - WHERE (example)

SELECT [First Names], Surname, Grade
FROM Students
WHERE Grade = 'A+'

“We will now together develop an SQL query that returns the first names, surname, and grade (in that order) of all students who have received an A+, with the results sorted by surname in alphabetical order.”
SQL queries – ORDER BY

Optional; allows us to sort our data in ascending or descending order

Syntax:
ORDER BY [name of field] [ASC/DESC]

SELECT [First Names], Surname, Grade
FROM Students
WHERE Grade = ‘A+’
ORDER BY Surname ASC

“We will now together develop an SQL query that returns the first names, surname, and grade (in that order) of all students who have received an A+, with the results sorted by surname in alphabetical order.”
SQL queries

You need to ensure that you put a semi-colon on the last clause of your SQL query:

```sql
SELECT [First Names], Surname, Grade
FROM Students
WHERE Grade = 'A+'
ORDER BY Surname ASC;
```
SQL queries

We run a SQL query in the same way that we run a QBE query

‘Run’ button
Write an SQL command that will (only) display the first names, surname and grade of students whose Total mark was greater than 70. Order the results table by ID number in ascending order.
SQL exercise

SELECT [First Names], Surname, Grade
FROM Students
WHERE Total > 70
ORDER BY ID ASC;
Retrieving information from a DB

Reports
Reports

Reports allow you to present the contents of a table, query etc. in a nicely formatted table.

There are two ways of creating Reports:

- Report Tool (show entire table, some formatting control)
- Report Wizard (table/field selection, grouping, sorting)
The Report Wizard

Select the tables and fields you want to display in your report
The Report Wizard

You can group records in the report using particular fields.
The Report Wizard

You can sort records in the report by one or more fields.
The Report Wizard

You can set certain aspects of your report’s formatting in the Wizard.

The final step involves giving the report a name and clicking on ‘Finish’.
The Report Wizard

The finished report, ready for printing

You can continue to modify the report’s formatting at this point
Thank you!