

9618 CAIE Computer Science — Databases

Alston

1 Introduction

Database management system (DBMS): responds to queries by extracting information from the database.

1.1 Terminology

Record: Storage representation of a row of data

Tuple: Specifically in a relational database, tuple is one row of data.

Field/Attribute: Field is a column, attribute is a characteristic that describes an entry in the database. (Attribute is used specifically for **DBMS**).

Primary key: An attribute in the table that uniquely identifies an entry in that table.

Foreign key: An attribute in a table that is the primary key of another table.

Compound key: Like a composite key (needs 2+ fields), but it's a combination of **foreign keys** in a table that uniquely identifies a row.

2 Relational Database vs File Based Systems

File based systems store all data in one singular table. This leads to **redundancies** in data, as the same thing might appear in multiple entries of the database. Thus this means there's a waste of space, as well as wasting time to edit these duplicate data.

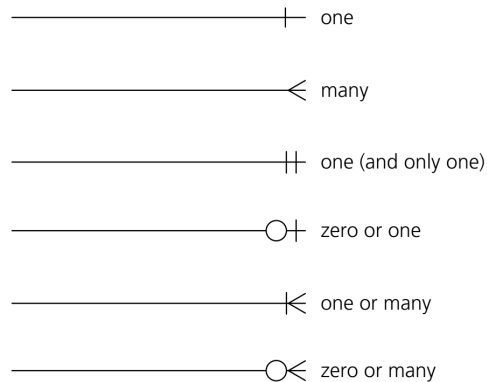
The solution is a relational database. Here there are MULTIPLE tables in the database, and they have relationships. These tables are linked by **primary keys** and **foreign keys**.

Theorem 2.1 – Benefits of relational database approach

- No wasted storage space
- Changed data is automatically reflected in another application
- Queries are not dependent on the structure of the data.

3 Relationships

There are many-to-many, many-to-one, one-to-many, and one-to-one relationships. An example: Consider a database with students and also information about classes. These two tables are linked by “classID” as a foreign key. There are many incidences of “classID” in the student table, but many students can only be matched to exactly one class. So the relationship is many-to-one. Draw the arrows to show the relationship in an ER diagram.



▲ **Figure 8.4** Cardinality of relationships

4 Normalisation

Ensures that database has integrity and that data redundancy is reduced.

4.1 First Normalisation (1NF)

Entities don't have repeated groups/attributes.

So you remove any repeated attributes from each table. Then you would create a separate table for these, and then link them with a foreign key.

4.2 Second Normalisation (2NF)

Any non-key attributes depend only on the primary key, there're no partial dependencies.

Partial dependencies happen when you're in a table and it has a composite primary key. But some of the attributes depend on only one specific part of the composite primary key, so for second normalisation you need to change this.

We just make another table by breaking up the composite key, and storing one of the broken up keys and the matching datas into that new table.

4.3 Third Normalisation (3NF)

All non-key values are independent. The table has no non-key dependencies.

So here, if in a table there are attributes that depend on another key that's not the primary key, just make a separate table.

5 DBMS

Data modelling and data dictionary: data dictionary stores meta data about the database (table names, columns in each table, datatypes they contain). Data modelling shows the structure of data: eg E-R diagram, and logical schemas. Logical schemas shows the tables and attributes of a database.

Use and purpose of DBMS:

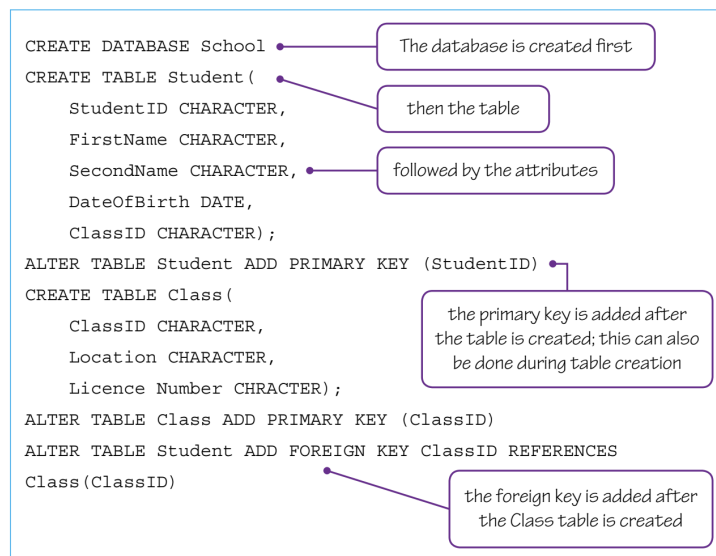
- Developer Interface: allows users to write queries, and these queries are processed by **query processor**.
- Query Processor: DDL processor and DML compiler.

6 DDL and DML

Data Definition Language (DDL): Creates datastructures like tables and attributes.

Data Manipulation Language (DML): Moves, modifies, deletes data in a database.

6.1 Data Definition Language



6.2 Data Manipulation Language

SQL (DML) query command	Description
SELECT FROM	Fetches data from a database. Queries always begin with SELECT.
WHERE	Includes only rows in a query that match a given condition
ORDER BY	Sorts the results from a query by a given column either alphabetically or numerically
GROUP BY	Arranges data into groups
INNER JOIN	Combines rows from different tables if the join condition is true
SUM	Returns the sum of all the values in the column
COUNT	Counts the number of rows where the column is not NUL
AVG	Returns the average value for a column with a numeric data type

SQL (DML) maintenance commands	Description
INSERT INTO	Adds new row(s) to a table
DELETE FROM	Removes row(s) from a table
UPDATE	Edits row(s) in a table

▲ Table 8.15 DML commands

This query will show, in alphabetical order of second name, the first and second names of all students in class 7A:

```
SELECT FirstName, SecondName
FROM Student
WHERE ClassID = '7A'
ORDER BY SecondName
```

This query will show the teacher's name and the subject taught:

```
SELECT Teacher.TeacherName AND Subject.SubjectName
FROM Teacher INNER JOIN Subject ON Teacher.
LicenceNumber = Subject.LicenceNumber
```

Note the INNER JOIN command in the examples: you're joining the columns from two distinct tables that have the same value. Syntax should be:

FROM Table1 INNER JOIN Table2 ON Table1.column1 = Table2.column2.

Also, when using "INSERT INTO", the syntax is INSERT INTO table1 VALUES(value1, value2...)