

RELATIONAL DATABASE

database in which the data items are linked by internal pointers

KEYS

- Candidate Key** = an attribute or smallest set of attributes in a table where no tuple has the same value
- Primary Key** = a unique identifier for a table
- Secondary Key** = a candidate key that is an alternative to the primary key
- Foreign Key** = a set of attributes in one table that refers to the primary key in another table

all attributes are candidate keys
↓
✓

Symbol	Name	Atomic Weight
H	Hydrogen	1.008
Li	Lithium	6.94
Na	Sodium	22.990

Symbol is the primary key
Name and Atomic Weight are secondary keys

▲ Table 8.3 Part of a table of elements

Relationship

formed when one table in a database has a foreign key that refers to a primary key in another table in the database

Student ID	First Name	Second Name	Date Of Birth	Class ID
S1276	Noor	Baig	09/22/2010	7A
S1277	Ahmed	Sayed	06/11/2010	7B
S1299	Tahir	Hassan	01/30/2011	7A

↑
Class ID is the foreign key

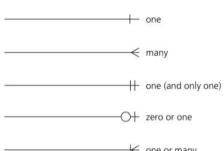
▲ Table 8.5 Part of a table for student entity

Class ID	Teacher Name	Location
7A	Mr Khan	Floor 2 Room 3
7B	Miss Malik	Floor 2 Room 4
7C	Miss Gill	Floor 2 Room 5

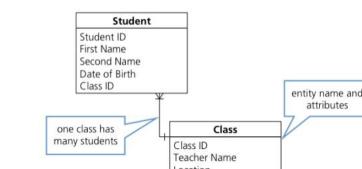
↑
Class ID is the primary key

▲ Table 8.6 Part of a table for class entity

Entity Relationship



An E-R diagram can be used to document the design of a database. This provides an easily understandable visual representation of how the entities in a database are related.



▲ Figure 8.3 E-R diagram for school database

NORMALISATION

used to construct a relational database that has integrity and in which data redundancy is reduced - table that has not been normalized appear larger

First Normal Form (1NF)

entities do not contain repeated groups of attributes

Student ID	First Name	Second Name	Date Of Birth	Class ID	Location	Teacher Name	Licence Number	Address	Teacher Date Of Birth
S1276	Noor	Baig	09/22/2010	7A	Floor 2 Room 3	Mr Khan	37952	School House 1	03/27/1985
S1277	Ahmed	Sayed	06/11/2010	7B	Floor 2 Room 4	Miss Malik	68943	School House 2	12/14/1988
S1299	Tahir	Hassan	01/30/2011	7A	Floor 2 Room 3	Mr Khan	37952	School House 1	03/27/1985

Student ID	Subject Name	Subject Teacher
S1276	Maths	Mr Yee
S1276	History	Miss Wu
S1276	Geography	Mr Khan
S1277	Maths	Mr Yee
S1277	Science	Miss Yo
S1277	Geography	Mr Khan
S1299	Maths	Mr Yee
S1299	Science	Miss Yo
S1299	History	Miss Wu

▲ Table 8.9 School database in 1NF

Second Normal Form (2NF)

entities are in 1NF and non-key attributes depend upon the primary key

Student ID	First Name	Second Name	Date Of Birth	Class ID	Location	Teacher Name	Licence Number	Address	Teacher Date Of Birth
S1276	Noor	Baig	09/22/2010	7A	Floor 2 Room 3	Mr Khan	37952	School House 1	03/27/1985
S1277	Ahmed	Sayed	06/11/2010	7B	Floor 2 Room 4	Miss Malik	68943	School House 2	12/14/1988
S1299	Tahir	Hassan	01/30/2011	7A	Floor 2 Room 3	Mr Khan	37952	School House 1	03/27/1985

Student ID	Subject Name
S1276	Maths
S1276	History
S1276	Geography
S1277	Maths
S1277	Science
S1277	Geography
S1299	Maths
S1299	Science
S1299	History

Subject Name	Subject Teacher
Maths	Mr Yee
History	Miss Wu
Geography	Mr Khan
Science	Miss Yo

▲ Table 8.10 School database in 2NF

Third Normal Form (3NF)

entities are in 2NF and all non-key attributes are independent - table contains no non-key dependencies

Student ID	First Name	Second Name	Date Of Birth	Class ID
S1276	Noor	Baig	09/22/2010	7A
S1277	Ahmed	Sayed	06/11/2010	7B
S1299	Tahir	Hassan	01/30/2011	7A

Licence Number	Teacher Name	Address	Teacher Date Of Birth
37952	Mr Khan	School House 1	03/27/1985
68943	Miss Malik	School House 2	12/14/1988
35859	Mr Yee	School House 1	10/07/1985
77248	Miss Yo	School House 2	05/05/1987
72691	Miss Wu	School House 2	11/21/1989
37952	Mr Khan	School House 1	03/27/1985

Class ID	Location	Licence Number
7A	Floor 2 Room 3	37952
7B	Floor 2 Room 4	68943

Student ID	Subject Name
S1276	Maths
S1276	History
S1276	Geography
S1277	Maths
S1277	Science
S1277	Geography
S1299	Maths
S1299	Science
S1299	History

▲ Table 8.11 School database in 3NF

DATABASE MANAGEMENT SYSTEMS (DBMS)

system software for the definition, creation, and manipulation of a database

DATA DEFINITION LANGUAGE (DDL)

a language to create, modify, and remove the data structures that form a database

DATA MANIPULATION LANGUAGE (DML)

a language used to add, modify, delete, and retrieve the data stored in a relational database

STRUCTURED QUERY LANGUAGE (SQL)

a list of SQL commands that perform a given task, often stored in a file for reuse

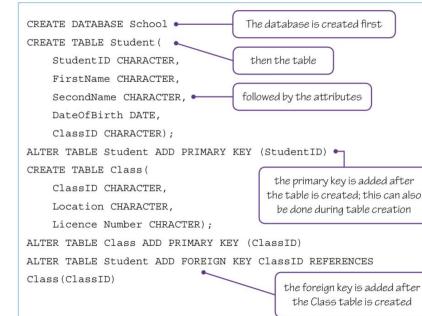
DDL commands

SQL (DDL) command	Description
CREATE DATABASE	Creates a database
CREATE TABLE	Creates a table definition
ALTER TABLE	Changes the definition of a table
PRIMARY KEY	Adds a primary key to a table
FOREIGN KEY ... REFERENCES ...	Adds a foreign key to a table

Data types for attributes

Data types for attributes	Description
CHARACTER	Fixed length text
VARCHAR(n)	Variable length text
BOOLEAN	True or False; SQL uses the integers 1 and 0
INTEGER	Whole number
REAL	Number with decimal places
DATE	A date usually formatted as YYYY-MM-DD
TIME	A time usually formatted as HH:MM:SS

Example DDL that could have been used when the school database was created



DML commands

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 NULL
AVG	Returns the average value for a column with a numeric data type

SQL (DML) maintenance commands

SQL (DML) maintenance command	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

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
```

This statement will insert a row into the Student table:

```
INSERT INTO Student VALUES(S1301, Peter, Probert,
06/06/2011, 7A)
```

If the values for all the columns are not known, then the table columns need to be specified before the values are inserted:

```
INSERT INTO Student(StudentID, FirstName, SecondName)
VALUES(S1301, Peter, Probert)
```

These statements will delete the specified row(s) from the Student table (take care: DELETE FROM Student will delete the whole table!):

```
DELETE FROM Student
WHERE StudentID = 'S1301'
```

The values for any column can be counted, totalled or averaged.

For example, if an extra column was added to the STUDENTSUBJECT table showing each student's exam mark in that subject, the following query could be used to total all of the students' exam marks:

```
SELECT SUM(ExamMark)
FROM STUDENTSUBJECT
```