

Structured Query Language

SQL

Schema Definition (DDL)



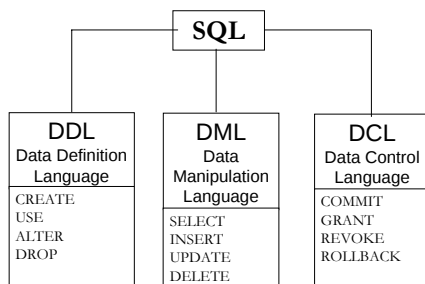
1

Introduction

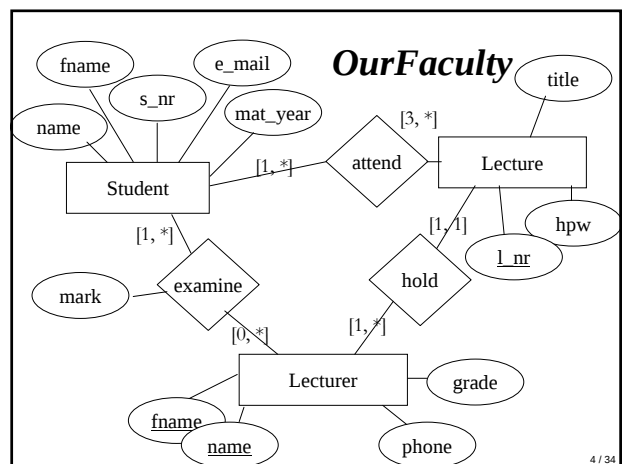
- 1974: first relational DBMS
- Based on Codd's relational model
- Current standard: SQL-99 (ANSI SQL-3)
- Available in all relational DBMS (but all SQL implementations differ from standard)
- Support of two interfaces
 - interactive: user friendly interface (MS Access)
 - application program interface: embedded SQL

2 / 34

Language elements



3 / 34



4 / 34

Example DB *OurFaculty*

Student(s_nr, name, fname, mat_year, e_mail)
 Lecturer(fname, name, grade, phone)
 Lecture(l_nr, title, hpw, fname, name)
 (fname, name) is foreign key
 examine(s_nr, fname, name, mark)
 attend(s_nr, l_nr)

5 / 34

Example DB *OurFaculty*

```

CREATE DATABASE IF NOT EXISTS OurFaculty;
USE OurFaculty;
CREATE TABLE Lecturer (
    fname VARCHAR(50) NOT NULL,
    name VARCHAR(50) NOT NULL,
    grade VARCHAR(30),
    phone VARCHAR(30),
    PRIMARY KEY (fname, name));
CREATE TABLE Lecture (
    l_nr INTEGER NOT NULL PRIMARY KEY,
    title VARCHAR(100) NOT NULL,
    hpw BYTE,
    fname VARCHAR(50),
    name VARCHAR(50),
    FOREIGN KEY (fname, name) REFERENCES
    Lecturer(fname, name));
    
```

6 / 34

CREATE table definition

```
CREATE TABLE <TableName> (  
  <attributeName> <attributeType>  
  [<constraint>]  
  
  [, <attributeName><attributeType>  
  [<constraint>]]  
  );
```

7 / 34

Some Integrity Constraints

Syntax: [CONSTRAINT [<name>]] <def>

Column constraints: NOT NULL
UNIQUE
CHECK

Cardinalities: PRIMARY KEY
FOREIGN KEY
REFERENCES

8 / 34

Basic Data Types

- Numbers
 - BYTE, INTEGER, LONG INTEGER, FLOAT, DOUBLE
- Characters, strings
 - CHARACTER, VARCHAR(n)
- Date and time
 - DATE (2007-09-25)
 - TIME (13:14:15)
- Binary objects
 - BIT(n) (010011)

9 / 34

Exercise 1

Continue the definition of *OurFaculty*

Student(s_nr, name, fname, mat_year, e_mail)
examine(s_nr, fname, name, mark)
attend(s_nr, l_nr)

10 / 34

A Solution

```
CREATE TABLE Student(  
  s_nr INTEGER PRIMARY KEY NOT NULL,  
  name VARCHAR(50) NOT NULL,  
  fname VARCHAR(50),  
  mat_year INTEGER,  
  e_mail VARCHAR(70));  
  
CREATE TABLE attend (  
  s_nr INTEGER,  
  l_nr INTEGER,  
  PRIMARY KEY (s_nr, l_nr),  
  FOREIGN KEY (s_nr)  
    REFERENCES Student(s_nr),  
  FOREIGN KEY (l_nr)  
    REFERENCES Lecture(l_nr) );
```

11 / 34

A Solution

```
CREATE TABLE examine (  
  s_nr INTEGER NOT NULL,  
  fname VARCHAR(50) NOT NULL,  
  name VARCHAR(50) NOT NULL,  
  mark BYTE,  
  PRIMARY KEY (s_nr, fname, name),  
  FOREIGN KEY (s_nr)  
    REFERENCES Student(s_nr),  
  FOREIGN KEY (fname, name)  
    REFERENCES Lecturer(fname, name));
```

12 / 34

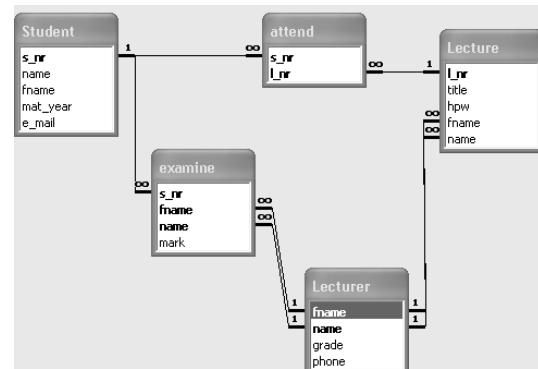
Exercise 2

- Start the Program *MS Access*
- Create the new database *OurFaculty.mdb*
- Create by SQL the tables *Lecturer*, *Student*, *Lecture*, *examine* and *attend* (as above described)

(Hint: for changing to SQL view click on the *Queries* tab in the Access main screen, click on the *New* button and select *Design* view, *Close* the appearing dialog box, and click the button in the top-left hand corner for switching to *SQL*)

13 / 34

Solution Relationships



14 / 34

Exercise 3

Define the DB *VideoStore*

Movie(m_id, category, title, year, director, length, prize_pd)
 Format(name, charge)
 Actor(stagename, realname, birthday)
 plays(stagename, m_id)
 Tape(t_id, name, m_id)

15 / 34

Solution Example *VideoStore*

```

CREATE TABLE Movie (
    m_id INTEGER PRIMARY KEY,
    title VARCHAR(60) NOT NULL,
    category CHAR(10),
    year DATE,
    director VARCHAR(30),
    price_pd DECIMAL(4,2),
    Length INTEGER,
    CONSTRAINT allowedPrice
    CHECK ( (price_pd >= 0) AND
    (price_pd < 100.0));
    
```

16 / 34

Solution Example *VideoStore*

```

CREATE TABLE Format(
    Name CHAR(5) PRIMARY KEY,
    Charge DECIMAL (3,2));

CREATE TABLE Tape(
    t_id INTEGER PRIMARY KEY,
    name CHAR(5) NOT NULL,
    m_id INTEGER NOT NULL,
    FOREIGN KEY (movie_id) REFERENCES
    Movie(id),
    FOREIGN KEY (format) REFERENCES
    Format(name));
    
```

17 / 34

Solution Example *VideoStore*

```

CREATE TABLE play (
    m_id INTEGER,
    stagename VARCHAR(30),
    PRIMARY KEY (m_id, stagename),
    FOREIGN KEY (m_id)
    REFERENCES Movie (m_id),
    FOREIGN KEY (stagename)
    REFERENCES Actor(stagename) );

CREATE TABLE Actor (
    stagename VARCHAR(30) NOT NULL UNIQUE,
    realname VARCHAR(30),
    birthday DATE );
    
```

18 / 34

ALTER table definitions

- **Syntax:**
`ALTER TABLE <tableName> <redefinition>`
- **Add an attribute:**
`ALTER TABLE Lecturer
ADD COLUMN e_mail VARCHAR(30);`
- **Change an attribute:**
`ALTER TABLE Lecturer
MODIFY e_mail VARCHAR(70);`

Remark: MS Access doesn't know the keyword MODIFY

19 / 34

DROP table definitions

- **Delete table only if not referenced:**
`DROP TABLE <tableName> restrict;`
- **Delete table and reference:**
`DROP TABLE <tableName> cascade;`

20 / 34

Exercise 4

- Start the Program *MS Access*
- Make a copy *OurFaculty2.mdb* of the database *OurFaculty.mdb* and open the copy
- Change the database *OurFaculty2.mdb* by altering the table *Lecturer* like above
- Try to delete in database *OurFaculty2.mdb* the table *Student* only if not referenced
- Delete in the database *OurFaculty2.mdb* the table *attend* and the reference

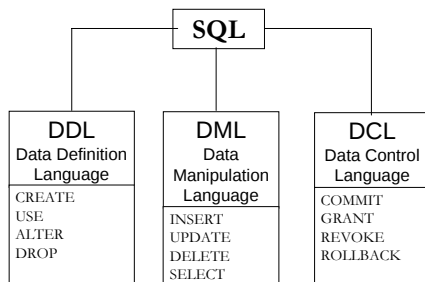
21 / 34

Structured Query Language SQL Data Manipulation Language (DML)



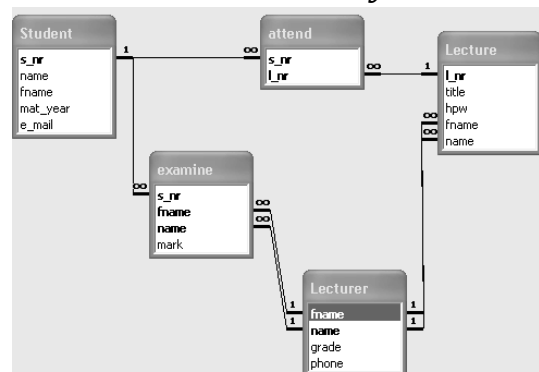
22

Remember language elements



23 / 34

Remember *OurFaculty.mdb*



24 / 34

INSERT data into tables

- Complete form:
Predefined order of values
INSERT INTO Student
VALUES (001, "Meyer", "Eve", 2005, NULL);
- Incomplete form:
Free order of values
INSERT INTO Student
(name, s_nr) **VALUES**
("Smith", 002);

25 / 34

DELETE data

- Syntax:
DELETE FROM <tableName>
[WHERE <predicate>];
- Example:
DELETE FROM Lecturer
WHERE fname = "Dietmar" **AND**
name = "Schmidt";
- Delete all rows:
DELETE FROM Student;

26 / 34

UPDATE data

- Syntax:
UPDATE <tableName>
SET <attr> = <value>
[, <attr> = <value>]
WHERE <predicate>;
- Example:
UPDATE Student
SET mat_year = 2005
WHERE s_nr = 4;

27 / 34

Exercise 1

- Start the Program *MS Access*
- Open the database *OurFaculty.mdb*
- Insert by SQL into the tables *Student*, *Lecture*, *Lecturer* and examine the following values

Student

	s_nr	name	fname	mat_year	e_mail
+	1	Meyer	Eve	2005	
+	2	Smith	Ryan	2005	r.smith@web.de
+	3	Orwell	Caroline	2006	
+	4	Hammer	Paul	2005	pauli@gmx.com

28 / 34

Exercise 1

Lecture

	l_nr	title	hpw	fname	name
	1	Databases	3	Heidrun	Koehler
	2	Discrete Mathematics	6	Dietmar	Cieslik
	3	Analysis	6	Werner	Schmidt
	4	Databases	4	Lutz	Voelkel

Lecturer

	fname	name	grade	phone	e_mail
▶	Dietmar	Cieslik	Professor	03834964600	cieslik@uni-greifswald.de
+	Dietmar	Schmidt	Doctor	03834967628	
+	Heidrun	Koehler	Pedagogue	03834964600	koehler@uni-greifswald.de
+	Lutz	Voelkel	Professor	03834967600	lv111@uni-greifswald.de
+	Werner	Schmidt	Professor	03834964629	wschmidt@uni-greifswald.de

examine

	s_nr	fname	name	mark
▶	1	Lutz	Voelkel	1,7
	3	Lutz	Voelkel	3,7
	4	Dietmar	Cieslik	2
	4	Heidrun	Koehler	1,3

29 / 34

Exercise 2

- Change by SQL the table *Student* so, that the student with the number 4 is also matriculated in 2006
- Delete the lecturer *Dietmar Schmidt*
- Insert again all values of the lecturer *Dietmar Schmidt*

30 / 34

SELECT data

- Basic Syntax:
SELECT [**DISTINCT**]<attr>
[, <attr>]
FROM <tableName>[, <tableName>]
[**WHERE** <predicate>]
[**ORDER BY** <attr>];
- Example:
SELECT fname, name, hpw
FROM Lecture
WHERE title = "Databases";

31 / 34

Exercise 3

Answer by SQL the questions:

- Who (first name and name) teaches *Databases* and how many hours a week?
- Which phone number has the lecturer *Heidrun Koehler*?
- Which students (name and first name) have got a mark better than 2?

32 / 34

Results

- **SELECT** fname, name, hpw
FROM Lecture
WHERE title = "Databases";
- **SELECT** phone
FROM Lecturer **WHERE** name = "Koehler" **AND**
fname = "Heidrun";
- **SELECT** mark, Student.name, Student.fname
FROM examine, Student
WHERE examine.s_nr = Student.s_nr
AND mark < 2
ORDER BY mark;

33 / 34

Further Examples

For further information about SQL see e. g. :

- http://www.itl.nist.gov/div897/ctg/dm/sql_examples.htm
- <http://www.1keydata.com/sql/sql.html>
- <http://www.soi.city.ac.uk/~tony/dbms/examples.html>

34 / 34