

БАЗИ ДАННИ

- "База данни е един вид колекция от организирани факти" (Ashenfelter, 1999).



1

БАЗИ ДАННИ

- СУБД е съвкупност от програми, която позволява създаването и поддържането на база данни.



2

БАЗИ ДАННИ

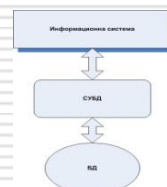
Функции на СУБД:

- Поддържа физическото и логическото ниво на съхранение на данните и механизмите на търсене;
- Осигурява средства за управление и поддръжка на БД;
- Осъществява достъп на потребители или приложения до базата данни за четене/запис;
- Реализира търсене на нужни на потребителя данни;
- Позволява поддържането на регламентиран достъп до данните.

3

ИНФОРМАЦИОННИ СИСТЕМИ

Информационната система е бизнес приложение. То се състои от БД, приложна програма и процедури по обработка на данните.

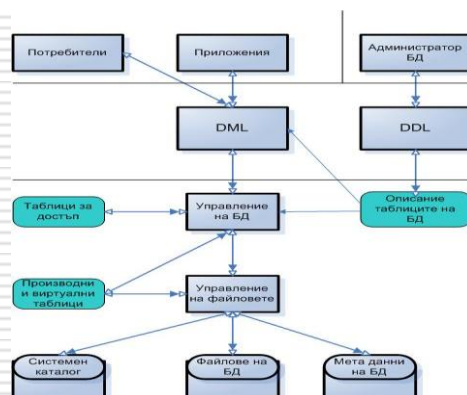


4

БАЗИ ДАННИ

- системни аналитици
- дизайнери на БД
- администратори на БД
- разработчици на приложения
- крайни потребители

5



6

SQL

SQL

- език за структурирани заявки към релационни БД;
- неизменна част от БД;
- непроцедурен език (какво, а не как), определя резултата, не стъпките;
- SQL е ANSI.

7

SQL

Първоначално 'Sequel' - Structured English QUery Language, който е част от проект на IBM през 70-те

SQL – Structured Query Language

Стандарти ANSI:

- SQL-89
- SQL-92 (SQL2)
- SQL-99 (SQL3)

8

SQL

SQL:

- може да се използва от други езици
- често се разширява, за да осигури програмни конструкции (ifthen, циклични структури, променливи и т.н.)

SQL предоставя:

- DDL
- DML
- DCL

9

БАЗИ ДАННИ

Функции на DDL

1. Дефинира характеристиките на всеки запис в базата данни: име и тип на данните за всяко поле.
2. Описание на схемата и подсхемите на базата данни.
3. Специфицира групирането на полетата в записите.
4. Идентифицира първичните и вторични ключове.
5. Осигурява връзките между отделните таблици и записи.
6. Управлява сигурността и интегритета, при които всеки запис има първичен ключ или при които зависимостта е минимизирана в релационната база данни.

10

БАЗИ ДАННИ

- Data Definition Language (DDL)

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- CREATE INDEX
- DROP INDEX

11

БАЗИ ДАННИ

Задачите, които изпълнява DML са:

1. Осигурява техники за управление на данните
2. Осигурява условия на работа с базата данни на високо ниво и лесни за разбиране термини.
3. Осигурява независимост на потребителя от физическата структура на данните.

12

БАЗИ ДАННИ

○ Data Manipulation Language (DML)

- SELECT
- UPDATE
- DELETE
- INSERT INTO

13

БАЗИ ДАННИ

SQL не е "case-sensitive" (обикновено);

Символните низове :

'STRING'

"It"s a STRING"

"

14

БАЗИ ДАННИ

SQL е непроцедурен език

Процедурен -

Непроцедурен -

Пример:

Student {ID, Name,Address}

Disciplines {Code, Title}

Choice {ID, Code}

Направи списък от студенти, които изучават 'Бази данни'

15

БАЗИ ДАННИ

First → първия запис на файла.	While (Student ≠ null)
Code=''	Choice → първи запис
While (First ≠ null) and (Code='')	While (Choice ≠ null)
If (First.Title= 'Бази данни') then	If (Choice.ID=Student.ID) And
Code=First.Code	(Choice.Code=Code) Then
First → следващ запис	Names=
Names → празен файл	Names+Student.Name
Student → първи запис	Choice → следващ запис
	Student → следващ запис
	Return Names

16

БАЗИ ДАННИ

SELECT Name FROM Student, Choice

WHERE

(Student.ID = Choice.ID)

AND

(Choice.Code =

(SELECT Code FROM Disciplines WHERE Title = 'Бази данни'))

17

БАЗИ ДАННИ

SQL е базиран на реляционния модел

- ✓ има същите идеи
- ✓ БД, които поддържат SQL, са често реляционни БД
- ✓ не винаги се придържа към модела

Е/Р проектирането може да се използва в SQL

- ✓ обекти, атрибути, релации се изразяват чрез SQL
- ✓ релацията много към много е проблем и трябва да се премахва

18

БАЗИ ДАННИ

Релации, обекти, таблици

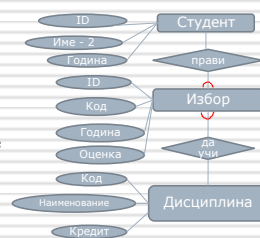
Релационен модел	E/R диаграми	SQL
Релация	Обект	Таблица
Кортеж	Инстанция	Ред
Атрибут	Атрибут	Колоната или поле
Първичен ключ		Първичен ключ
Чужд ключ	M:1 релация	Чужд ключ

19

ER диаграми

E/R диаграми → таблици

- Всеки обект – таблица
- Атрибутите стават колони
- Отношенията могат да се представят и с чужди ключове



доц. Стоянова

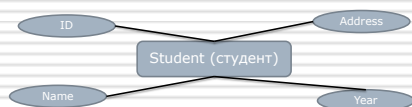
20

ER диаграми

Обекти и атрибути

Всеки обект става таблица от базата данни

- Име на таблицата – името на обекта
- Атрибутите – колони на таблицата със същото име
- Таблица Student
- С колони за ID, Name, Address, Year



CREATE TABLE

CREATE TABLE <name> (<col-def-1>, <col-def-2>, ... <col-def-n>, <constraint-1>, ... <constraint-k>)

Определят се:

- Име на таблицата
- Списък от дефиниции на колони
- Списък от ограничения (напр. ключове)

22

БАЗИ ДАННИ

<col-name> <type> [NULL|NOT NULL] [DEFAULT <val>] [constraint-1 [, constraint-2[, ...]]]

Всяка колона има име и тип

Общи типове:

- INT
- REAL
- CHAR(n) NCHAR(n)
- VARCHAR(n) NVARCHAR(n)
- DATE

23

БАЗИ ДАННИ

- За колоната трябва да се зададе **NULL** или **NOT NULL**
- Колоните **NOT NULL** не могат да бъдат без стойност
- Ако не се упомене в дефиницията - **NULL**
- На колона може да се зададе стойност по подразбиране
- Ключовата дума е **DEFAULT**, напр.
num INT DEFAULT 0

24

БАЗИ ДАННИ

```
CREATE TABLE Student (sID INT NOT NULL, sName VARCHAR(50) NOT NULL, sAddress VARCHAR(50), sYear INT DEFAULT 1)
```



25

БАЗИ ДАННИ

Ограничения (constraints) ➤

CONSTRAINT

<name>

<type>

<details>

Общи типове:

- **PRIMARY KEY**
- **UNIQUE**
- **FOREIGN KEY**
- **INDEX**

➤ Всяко ограничение получава име – Access

➤ Ограниченията за една колона могат да се включат в дефиницията

26

БАЗИ ДАННИ

- Ограничения за дефиниране на първичен ключ
- Ограничението **PRIMARY KEY** включва ограничението **UNIQUE** и също така включва ограничението **NOT NULL**
- **<details>** на първичен ключ – списък на колоните

```
CONSTRAINT <name> PRIMARY KEY (col1, col2, ...)
```

27

БАЗИ ДАННИ

- Освен първичния ключ и набор от други колони може да бъде **UNIQUE**
- Ефектът – кандидат ключове
- **<details>** – списък на колоните на кандидат ключовете

```
CONSTRAINT <name> UNIQUE (col1, col2, ...)
```

28

БАЗИ ДАННИ

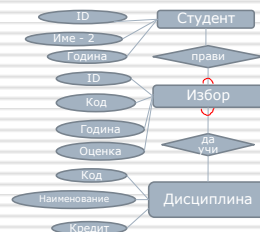
```
CREATE TABLE Student (sID INT NOT NULL, sName VARCHAR(50) NOT NULL, sAddress VARCHAR(50), sYear INT DEFAULT 1, CONSTRAINT pkStudent PRIMARY KEY (sID))
```

29

ER диаграми

Релациите

- 1:1 обикновено не се използват, с тях се работи като с M:1
- M:1 се представят като чужд ключ от страна M към страна 1
- M:M се представя като две M:1 релации



доц. Стоянова

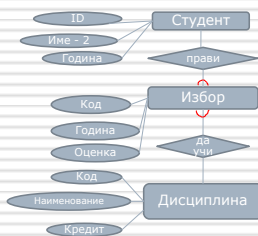
30

ER диаграми

Релациите в таблицата

Choice

- Ще има колони за посочените атрибути
- Ще има чужд ключ към Student за релацията 'прави'
- Ще има чужд ключ към Discipline за релацията 'да учи'



доц. Стоянова

31

БАЗИ ДАННИ

Чуждите ключове (FK) се дефинират като ограничения (constraints)

Дефинират се:

- Колоните, които формират FK
- Рефрираната таблица
- Колоните, които се рефрират от FK

CONSTRAINT <name> FOREIGN KEY (col1,col2,...)

REFERENCES <table> [(ref1,ref2,...)]

- Ако FK рефрира PK на <table> не е необходимо да се изброяват колоните

32

БАЗИ ДАННИ

```
CREATE TABLE Choice (sID INT NOT NULL, dCode CHAR(6)
NOT NULL, cYear INT, cExam INT, CONSTRAINT cPK
PRIMARY KEY (sID, dCode), CONSTRAINT cStu FOREIGN
KEY (sID) REFERENCES Student (sID), CONSTRAINT cCode
FOREIGN KEY (dCode) REFERENCES Disciplines (dCode))
```

33

БАЗИ ДАННИ

За тази лекция:

- Какво е непроцедурен език? Сравнете с процедурен.
- Що е SQL, DDL, DML?
- Създаване на таблици от описание на изискванията или от ER диаграма.

доц. Стоянова

34



доц. Стоянова

35