

Database Design  
kurs 2017 – database design and programming with sql  
kombinacija teachers i students slajdova

9-1 Uvod u koncepte relacionih baza podataka

Smisao

- Konceptualni data model će se transformisati u relacioni dbd; to znači da naši entiteti, atributi, relacije i UID će se transformisati u objekte u relacionoj db
- Transformisati će se (mutating) jedan set objekata (ER konceptualnog modelovanog konstrukta) u drugi (data design fizički model konstrukt)
- Ilustracija relacione db
- Relaciona db je db koju korisnik vidi kao kolekciju 2D tabela, svaka sa redovima i kolonama

EMPLOYEES (table name)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
200	Jennifer	Whalen	10
205	Shelley	Higgins	110

Row

Column

- Relaciona db: kolekcija objekata ili relacija, skup operatora koji reaguju nad tim relacijama i podataka o integritetu preciznosti i konzistentnosti.
- Svaki red podataka opisuje zaposlenog. Svaka kolona je atribut tog zaposlenog. Ako bismo želeli pronaći prezime i broj sektora zaposlenog broj 210, treba da pristupimo trećem redu tabele a zatim da pronađemo vrednosti za first\_name i departmant\_no za taj red. Ali kako da pronađemo dobar red pre svega ? Treba li da idemo kroz celu tabelu i pogledamo u svaki red ?

Jezik pristupa podacima

- SQL (Structured query language) nam omogućava pristup podacima u relacionim db na efikasan način
- Umesto da ručno pretražujemo preko svakog reda za pronaći zapis o zaposlenom broj 200, koristimo sledeći SQL iskaz:

EMPLOYEES (table name)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
200	Jennifer	Whalen	10
205	Shelley	Higgins	110

```
SELECT last_name, department_id
FROM employees
WHERE employee_id = 200;
```

LAST_NAME	DEPARTMENT_ID
Whalen	10

## Poseban SQL upit

- Za pronalazak svih zaposlenih u sektoru broj 90:

EMPLOYEES (table name)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	DEPARTMENT_ID
100	Steven	King	...	90
101	Neena	Kochhar	...	90
102	Lex	De Haan	...	90
200	Jennifer	Whalen	...	10
205	Shelley	Higgins	...	110

```
SELECT *  
FROM employees  
WHERE department_id = 90;
```

EMPLOYEES (table name)

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	DEPARTMENT_ID
100	Steven	King	...	90
101	Neena	Kochhar	...	90
102	Lex	De Haan	...	90

## Osnovni ključ

- Osnovni ključ (PK) je kolona ili skup kolona koje jedinstveno identifikuju svaki red u tabeli
- Osnovni ključ : Ograničenje koje osigurava da kolona ne sadrži null vrednost i jedinstveno identifikuje svaki red u tabeli
- Videti da u ACCOUNTS, BANK\_NO nije jedinstveno i ACCT\_NO nije jedinstveno; ipak kombinacija BANK\_NO i ACCT\_NO jeste jedinstveno

ACCOUNTS

BANK_NO	ACCT_NO	BALANCE	DATE_OPENED
104	75760	12,0050.00	21-OCT-89
104	77956	100.10	
105	89570	55,775.00	15-JAN-85
103	55890	15,001.85	10-MAR-91
105	75760	5.00	22-SEP-03

Multiple Column Primary Key

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	DEPARTMENT_ID
100	Steven	King	...	90
101	Neena	Kochhar	...	90
102	Lex	De Haan	...	90
200	Jennifer	Whalen	...	10
205	Shelley	Higgins	...	110

Single Column Primary Key

- Nijedan deo primarnog ključa ne može biti null
- ## Kandidati za osnovni ključ
- Tabela može imati više od jedne kolone ili kombinacije kolona koje mogu poslužiti kao osnovni ključ
  - Svaka kolona ili kombinacija kolona se naziva kandidat ključ jer može biti izabrana kao osnovni ključ

MEMBERS

MEMBER_ID	LAST_NAME	FIRST_NAME	PAYROLL_ID
100	SMITH	DANA	21215
310	ADAMS	TYLER	59877
210	CHEN	LAWRENCE	1101
405	GOMEZ	CARLOS	52
378	LOUNGANI	NEIL	90386

↑ Candidate Key

Candidate Key ↑

- Šta čini EMPLOYEE\_ID i AYROLL\_ID dobre kandidate za osnovni ključ ? Oba su jedinstvena i nisu null.

Izbor kandidata za ključ

- Izaberi jednog kandidata za osnovni ključ; ostali kandidati postaju alternate (alternativni) ključevi ili jedinstveni (unique) ključevi

MEMBERS

MEMBER_ID	LAST_NAME	FIRST_NAME	PAYROLL_ID
100	SMITH	DANA	21215
310	ADAMS	TYLER	59877
210	CHEN	LAWRENCE	1101
405	GOMEZ	CARLOS	52
378	LOUNGANI	NEIL	90386

↑  
Primary Key

↑  
Alternate or  
Unique Key (UK)

- Unique key: Integrity ograničenje koje zahteva da svaka vrednost u koloni ili setu kolona bude jedinstvena
- Zašto je korisno imati alternativan ili jedinstven ključ ? To je drugi način za lociranje zapisa. Ako zaboravite vaš employee ID, ali znate payroll ID, možete i dalje pristupiti zapisu zaposlenog.

Strani (foreign) key

- Strani ključ (FK) je kolona ili kombinacija kolona u jednoj tabeli koja sadrži vrednosti koje odgovaraju primarnom ključu u drugoj tabeli
- Ako je FK u EMPLOYEES ima vrednost 10, onda mora postojati red u DEPARTMANTS sa DEPARTMENT\_ID od 10. Drugačije, to je prekršaj referencijalnog integriteta

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
200	Jennifer	Whalen	10
205	Shelley	Higgins	110

Foreign Key ↓

DEPARTMENTS

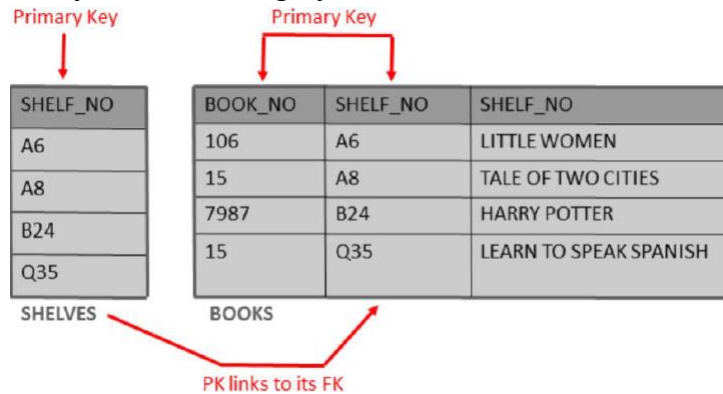
DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
50	Shipping

↑ Primary Key

refers to

### Pravila stranog ključa

- Ako je osnovni ključ sastavljen od jednog ili više stranih ključeva, FK vrednost ne sme biti NUL
- U ovom primeru, SHELF\_NO je deo osnovnog ključa od BOOKS. Takođe je FK za SHELVES. Pošto je deo osnovnog ključa od BOOKS, ne može biti null



- Integritet kolona
- Kolona mora sadržati samo vrednosti koje su u skladu sa definisanim formatima podataka u koloni
- Red: ulaz u tabeli, koji se sastoji od vrednosti za svaku odgovarajuću kolone
- Kolona: implementacija atributa ili relacije u tabeli

### ACCOUNTS

BANK_NO	ACCT_NO	BALANCE	DATE_OPENED
104	75760	12,0050.00	21-OCT-1989
104	77956	100.10	
105	89570	55,775.00	15-JAN-1985
103	55890	15,001.85	10-MAR-1991
105	75760	5.00	22-SEP-2003

### ACCOUNTS Table Definition

Column Name	Data Type	Optionality
BANK_NO	Number (5)	Not null
ACCT_NO	Number (8)	Not null
BALANCE	Number (12,2)	Not null
DATE_OPENED	Date	

### Suma Data-Integrity pravila (DIR)

- DIR (constraints) definišu relacionu ispravnost stanja db; omogućavaju da korisnici mogu izvesti samo one operacije koje ostavljaju db u dobrom, odgovarajućem stanju

Constraint Type	Explanation	Example
Entity Integrity	A primary key must be unique, and no part of the primary key can be null	The column emp_no in the EMPLOYEES table cannot be null
Referential Integrity	A foreign key must match an existing primary key value (or else be null if nulls are allowed)	The value in the dept_no column of the EMPLOYEES table must match a value in the dept_no column in the DEPARTMENTS table
Column Integrity	A column must contain only values consistent with the defined data format of the column	The value in the balance column of the ACCOUNTS table must be numeric
User-Defined Integrity	The data stored in a database must comply with the rules of the business	If the value in the balance column of the ACCOUNTS table is below 1.00, we must send a letter to the account owner (this will need additional programming to enforce)