

Database Programming with SQL
kurs 2017 – database design and programming with sql
students slajdovi

Osnovni SELECT iskazi

- SELECT * komanda vraća sve redove u tabeli
- Sintaksa je (pravila o formacijama iskaza u programskom jeziku):
SELECT *
FROM <table name>;
- Npr:
SELECT *
FROM employees;

SELECT iskazi sa Condition

- Za vraćanje subseta podataka, modifikuj SELECT iskaz:

```
SELECT <column name 1, column name 2, etc.>  
FROM <table name>  
WHERE <condition>;
```

- Npr,
- <condition> počinje sa reči WHERE koju prati <column name> i Comparison Operator (=, <, >...) kojeg prati vrednost ili IS NULL, IS NOT NULL
- Subset: deo veće grupe povezanih stvari
- Comparison Operator: koristi u uslovima koji upoređuju jedan izraz sa drugim izrazom ili vrednosti

Correcting errors

- Prilikom unosa SQL komandi, važno je koristiti pravilno spelovanje, ili će se javiti greška
- Npr,
SEECT *
FROM SELECT first_name, last_name, job_id
 FROM employees employees;
će dati WHERE job_id = 'SA_REP'; poruku o grešci: ORA-00900:
invalid SQL statement
- Takođe je bitno koristiti pravilna imena i spelovanje za kolone i tabele
- Npr,
SELECT *
FROM employee;
će dati poruku o grešci; ORA-00942: table or view does not exist
- Npr, ako je ime kolone nepravilno treba first_name
SELECT name
FROM employees;
će dati poruku o grešci: ORA-00904: "NAME": invalid identifier

1-1 Relational Database Technology

- Mnogi gradovi koriste inteligentan sistem uređivanja saobraćaja sa db za kontrolu semafora

Relational Databases

- Relacione db omogućavaju tabelama da budu međusobno u relaciji preko zajedničkih polja
- Relaciona db: Kolekcije objekata ili relacija, set operatora koji deluju preko tih relacija i data integrity za preciznost i konzistentnost
- Polje: intersekcija reda i kolone
- U realnosti, db u biznisu imaju puno tabela, svaka tabela deli zajedničko polje sa drugom tabelom
- Tabela "countries" je jedna od nekoliko tabela u Employees db i samo jedan primer od mnogo tabela koje će se koristiti
- Kolona "Region_ID" je zajedničko polje sa "regions" tabelom
- RDBMS: Relational Database Management System
- Oracle RDBMS koriste: France Telecom, Amazon.com, The Claria Corporation

COUNTRY_ID	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

Review Key Terms

- **table** (basic storage structure), **column** (one kind of data in a table), **row** (data for one table instance), **field** (the one value found at the intersection of a row and a column), **primary key** (unique identifier for each row), **foreign key** (column that refers to a primary-key column in another table)

Properties of Tables

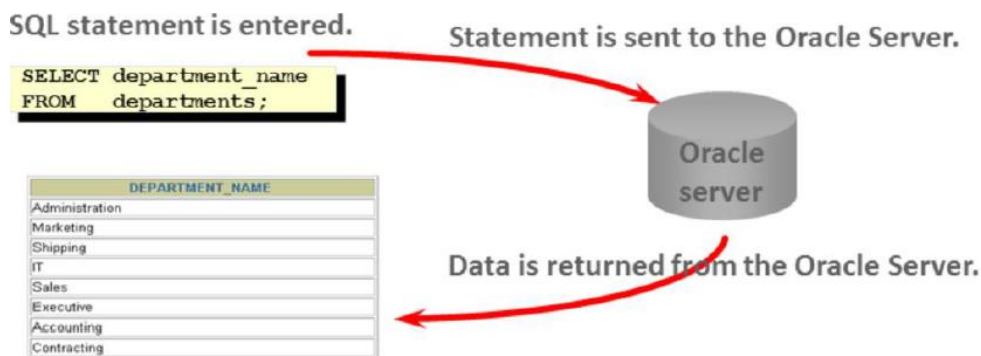
- Postoji šest osobina tabela u relacionim db:
 1. ulazi u kolonama su sa jednom vrednosti (single-valued)
 2. ulazi u kolonama su iste vrste
 3. svaki red je jedinstven
 4. sekvenca kolona je nevažna
 5. sekvenca redova je nevažna
 6. svaka kolona ima jedinstveno ime

Accessing Data in an RDBMS

- RDBMS organizuje podatke u povezane redove i kolone; za pristup podacima ne treba znati gde je podatak fizički smešten niti kako se pristupa tabeli; korišćenjem SQL (structured query language) iskaza i operatora se pristupa podacima

Komunikacija sa db

- Rad sa db: prvo se izabere metod komunikacije; po uspostavljanju komunikacije postavi se pitanje (query); na postavljeno pitanje dobija se odgovor (podatak)
- Komunikacija sa RDBMS je preko SQL iskaza unutar APEX



- Zahtev se onda šalje Oracle Serveru (db koja radi na računaru), zahtev se obradi a vraćeni podatak se prikaže
- U velikim db sistemima, korisnici, serveri i tabelle čine RDBMS

Kategorije SQL iskaza

- SQL iskazi se grupišu u nekoliko kategorija u zavisnosti od funkcije koju realizuju
- Iskazi koji vraćaju podatak iz db koriste rezervisanu reč SELECT; ona je limitiran oblik DML iskaza pošto ona može pristupiti samo podacima u db; ne može manipulirati podacima u db, iako može raditi na pristupnim podacima pre vraćanja rezultata upita
- Postoje četiri kategorije SQL iskaza: Data manipulation language (DML), Data definition language (DDL), Transaction control language (TCL), Data control language (DCL)
- **DML** iskazi počinju sa INSERT, UPDATE, DELETE ili MERGE i koriste se za modifikovanje podataka u tabeli unosom novih redova, izmenom postojećih redova ili odstranjivanjem postojećih redova
- **DDL** iskazi kreiraju, menjaju i odstranjuju strukture podataka iz db; koriste se CREATE, ALTER, RENAME i TRUNCATE
- **TCL** iskazi se koriste za rad sa promenama koje su realizovane sa DML iskazima; promene nad podacima se izvršavaju sa COMMIT, ROLLBACK i SAVEPOINT; ove promene se mogu grupisati u logičke transakcije
- **DCL** službene reči GRANT i REVOKE se koriste za davanje ili odstranjivanje prava pristupa u db i strukturama unutar nje

1-2 Anatomy of a SQL Statement

SELECT službena reč

- SELECT se koristi za vraćanje informacija iz db
- Neka db ima podatke o filmovima kao što su title, ganre, studio, producer, release date, series, country, language, rating, running time...
- Šta ako tražimo samo naslove filmova napravljenih u Indiji ? Za to se koristi SELECT.
- Sintaksa SELECT iskaza:

```
SELECT <column_name(s)>
FROM <table_name>;
```

- U najjednostavnijoj formi, SELECT iskaz mora uključiti sledeće: SELECT rečenicu koja specificira kolone koje treba prikazati; FROM rečenicu koja specificira tabelu koja sadrži kolone izlistane u SELECT rečenici
- SELECT rečenica: specificira kolone za prikaz (ili koristi * simbol za prikaz svih kolona)
- FROM rečenica: specificira tabelu sa kolonama izlistanim u SELECT rečenici

- Column: implementacija atributa ili relacija u tabeli

Konvencije

- Koristiće se:
- Službena reč (**keyword**) se odnosi na pojedinačnu SQL komandu (SELECT i FROM su službene reči); rečenica (**clause**) je deo SQL iskaza (SELECT last_name je rečenica); iskaz (**statement**) je kombinacija dve ili više rečenica (SELECT last_name FROM employees je iskaz)
- SQL službene reči se pišu velikim slovima, nazivi kolona i tabela se pišu malim slovima, svaka rečenica se piše u novom redu:

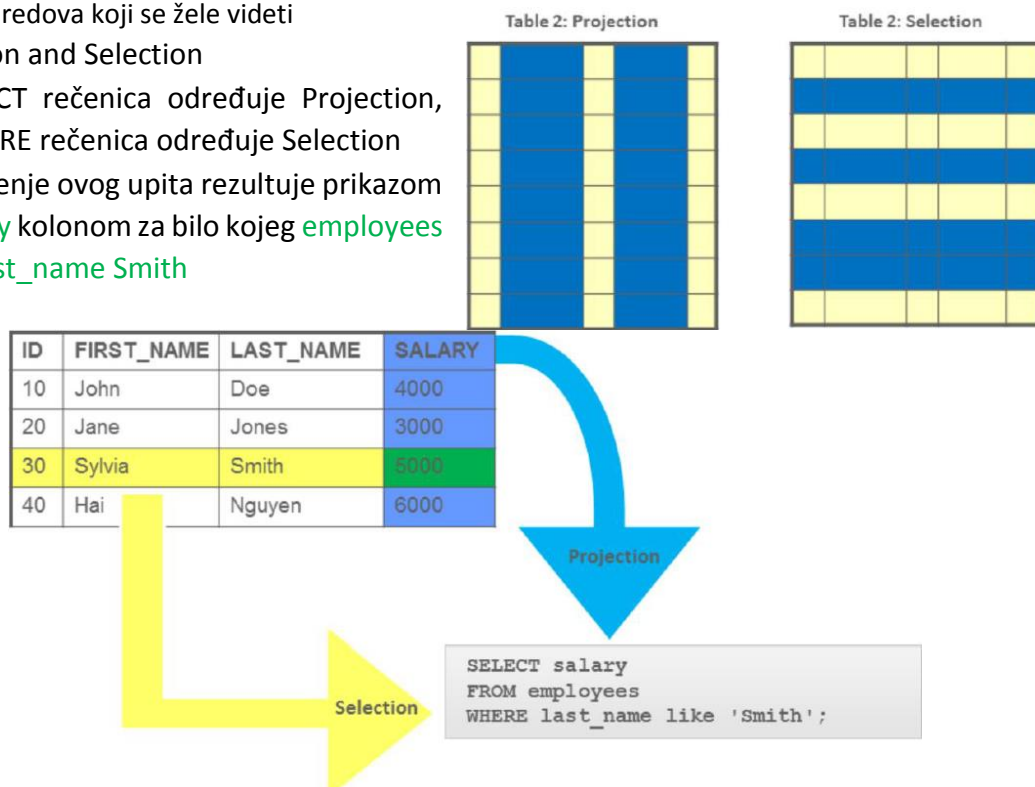
```
SELECT last_name
FROM employees
WHERE employee_id = 101;
```

Capabilities of SELECT Statements

- Projection: koristi se za izbor kolone u tabeli; mogu se koristiti projection mogućnosti u SQL za izbor kolona u tabeli koje želimo vratiti preko upita; može se izabrati bilo koji broj kolona iz tabele
- Selection: koristi se za izbor redova u tabeli; može se koristiti selection mogućnosti u SQL za izbor redova iz tabele koje želimo vratiti preko upita; mogu se koristiti različiti kriterijumi za izbor redova koji se žele videti

Projection and Selection

- SELECT rečenica određuje Projection, WHERE rečenica određuje Selection
- Izvršenje ovog upita rezultuje prikazom salary kolonom za bilo kojeg employees sa last_name Smith



Izbor svih kolona

- Mogu se prikazati sve kolone sa podacima u tabeli korišćenjem asterisk simbola (*) umesto imena kolone u SELECT rečenici
- Ponekad pisanjem **SELECT * FROM table** zahteva dosta vremena za prikaz podataka; sve zavisi od broja redova smeštenih u traženoj tabeli

- Drugi način prikaza svi kolona je individualnim navođenjem imena svake kolone:
- Zarezi posle navođenja imena nisu obavezni za APEX ali pošto drugi Oracle SQL Interfaces to zahtevaju, onda se ovde i navedene

Projecting Specific Columns

- Ako se želi to PROJECT samo specifične kolone iz tabele da bi se prikazale, samo je potrebno napisati imena svake kolone razdvojene zarezima unutar SELECT rečenice

```
SELECT country_id, country_name, region_id
FROM countries;
```

COUNTRY_ID	COUNTRY_NAME	REGION_ID
CA	Canada	2
DE	Germany	1
UK	United Kingdom	1
US	United States of America	2

```
SELECT location_id, city, state_province
FROM locations;
```

LOCATION_ID	CITY	STATE_PROVINCE
1800	Toronto	Ontario
2500	Oxford	Oxford
1400	Southlake	Texas
1500	South San Francisco	California
1700	Seattle	Washington

Using Arithmetic Operators

- Ako je potrebno modifikovati način na koji se podaci prikazuju, izvode kalkulacije ili pregledati šta ako scenarije (npr, Šta ako je svakom zaposlenom dato 5% povišica ?)
- Ovakvi tipovi kalkulacija su mogući korišćenjem aritmetičkih izraza (add (+), subtract (-), multiply (*), divide (/))
- Rezultati ovakvih kalkulacija će se pojaviti samo na izlazu i neće uticati na podatke u tabelama
- Primer pokazuje operator sabiranja za računanje salry increase od 300 za sve zaposlene i prikazuje novi SALARY + 300 kolonu na izlazu
- Postavljanje praznih mesta pre i posle aritmetičkog operatora neće uticati na izlaz

Precedence in Arithmetic Operators

- Prioritet (precedence) je redosled u kojem Oracle

```
SELECT last_name, salary, salary + 300
FROM employees;
```

preračunava (evaluate) različite operatore u istom izrazu

- Pri preračunavanju izraza koji sadrže višestruke operatore, Oracle prvo preračunava operatore sa visokim prioritetom pa onda sa niskim prioritetom, dok operatore sa istim prioritetom preračunava sa leva na desno unutar izraza
- Redosled prioriteta je: * / + - (pri čemu su množenje i deljenje istog prioriteta)

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Whalen	4400	4700
Higgins	12000	12300
Gietz	8300	8600
Zlotkey	10500	10800
Abel	11000	11300
Taylor	8600	8900
Grant	7000	7300

Operator Precedence

```
SELECT last_name, salary,  
       12*salary +100  
FROM employees;
```

Using Parentheses

```
SELECT last_name, salary,  
       12*(salary +100)  
FROM employees;
```

- Primer sa leve strane koristi operatorske prioritete za dobijanje rezultata, primer sa desne strane prvo koristi operaciju u zagradi a zatim *

NULL Value

- NULL je vrednost koja je nedostupna (unavailable), nedodeljena (unassigned), nepoznata (unknown) ili neprimenjiva (inapplicable)
- NULL nije isto što i nula ili prazno mesto; u SQL nula je vrednost, prazno mesto je karakter
- Ponekad se ne zna vrednost za kolonu
- U db može se smestiti nepoznato
- Relaciona db koristi držač (placeholder) zvan NULL ili null za predstavljanje ovih nepoznatih vrednosti
- Ako je bilo koja kolona u aritmetičkom izrazu null, rezultat je null ili nepoznat
- Ako se pokuša deliti sa null, rezultat je null ili nepoznat
- Ipak ako se deli sa nula, dobija se greška

Salaries and Commissions

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	-
Kochhar	AD_VP	17000	-
De Haan	AD_VP	17000	-
Whalen	AD_ASST	4400	-
Higgins	AC_MGR	12000	-
Gietz	AC_ACCOUNT	8300	-
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3

```
SELECT last_name, job_id, salary, commission_pct, salary*commission_pct  
FROM employees;
```

Salaries and Commissions

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT	SALARY*COMMISSION_PCT
King	AD_PRES	24000	-	-
Kochhar	AD_VP	17000	-	-
De Haan	AD_VP	17000	-	-
Whalen	AD_ASST	4400	-	-
Higgins	AC_MGR	12000	-	-
Gietz	AC_ACCOUNT	8300	-	-
Zlotkey	SA_MAN	10500	.2	2100
Abel	SA_REP	11000	.3	3300
Taylor	SA_REP	8600	.2	1720

- U primeru, red sa null vrednosti u commission_pct koloni će vratiti null vrednost za salary*commission_pct kolonu u ovom upitu
- NULL vrednosti se u APEX predstavljaju sa – (dash) simbolom

Aliases

- Alijas je način reimenovanja naslova kolone na izlazu
- Bez alijasa, kada je rezultat SQL iskaza prikazan, ime prikazanih kolona će biti isto kao imena kolona u tabeli ili ime prikazano za prikaz aritmetičke operacije kao što je 12*(SALARY + 100)
- Alijasi kolona omogućavaju reimenovanje kolona na izlazu
- Postoji nekoliko pravila pri korišćenju alijasa kolona za formatiranje izlaza:
 - alijas reimenuje naslov kolone (ne menja pravi naziv kolone)
 - korisno je pri kalkulacijama

- odmah prati ime kolone
- može imati opcionalnu AS službenu reč između imena kolone i alijasa
- traži dvostruke navodnike ako alijas sadrži prazna mesta ili specijalne karaktere, ili je osetljivo na veličinu slova

Using Column Aliases

- Sintaksa za alijase je:

```
SELECT * |column|expr [ AS alias], .....
FROM      table;
```

- Examples:

```
SELECT last_name AS name,
       commission_pct AS comm
FROM employees;
```

NAME	COMM
King	-
Kochhar	-
De Haan	-

```
SELECT last_name "Name",
       salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

- Dvostruki navodnici su neophodni za alijase duže od jedne reči ili kada je format alijasa bilo šta drugo sem difolt uppercase imena (npr Department Name ili department ili dept.)
- Prvi primer koristi AS službenu reč, ali nema dvostruke navodnike, tako da alijasi su prikazani u difolt upper slovima
- Drugi primer ne koristi AS, ali iako je namenjeni alijas duži od jedne reči, koristi dvostruke navodnike
- Korišćenje opcionalnih AS čini SQL iskaze lakše za čitanje