

### Svojstva virtualne relacije

Obavljanjem naredbe CREATE VIEW u rječnik podataka se pohranjuje samo definicija virtualne relacije

- sadržaj virtualne relacije se određuje tek za vrijeme izvršavanja upita koji koristi virtualnu relaciju

- odnosno, sadržaj virtualne relacije uvijek odražava sadržaj temeljnih relacija u trenutku izvršavanja upita u kojem se virtualna relacija koristi

Virtualne relacije se u upitima mogu koristiti na svim mjestima gdje se mogu koristiti temeljne relacije

- između ostalog i za kreiranje novih virtualnih relacija

Za razliku od privremene relacije

- definicija virtualne relacije je trajno pohranjena u bazi podataka
- virtualna relacija je u dosegu ("vidljiva je") u svim SQL-sjednicama

### Kako sustavi za upravljanje bazama podataka izvršavaju upite koji sadrže virtualne relacije?

- modifikacijom upita
- korištenjem materijalizirane virtualne relacije

### Izmjenjive virtualne relacije

Virtualna relacija je izmjenjiva ukoliko u glavnom SELECT dijelu definicije virtualne

relacije koristi attribute iz samo jedne temeljne relacije  $r(R)$  i pri tome ne sadrži:

- eliminaciju duplikata pomoću DISTINCT
- izraze u listi za selekciju (osim izraza koji sadrže samo ime atributa)
- spajanje ili uniju
- grupiranje i postavljanje uvjeta nad grupom (GROUP BY i HAVING)

Prethodno navedena ograničenja se ne odnose na eventualne podupite koji se koriste

- unutar glavnog SELECT dijela definicije virtualne relacije, ali
- podupiti ne smiju u svojem FROM dijelu koristiti relaciju  $r(R)$

### Entitet

Bilo što, što ima suštinu ili bit, ima jasnoću kao činjenica ili ideja, posjeduje značajke s pomoću kojih se može razlučiti od svoje okoline

#### Skup entiteta $E$ (*entityset*)

Slični entiteti se grupiraju u skupove entiteta

#### Skup veza $R$ (*relationship set*)

- matematička relacija između  $n$  entiteta:

$$R \subseteq E_1 \times E_2 \times E_3 \times \dots \times E_n$$

$$\text{ili } R = \{ (e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n \}$$

$n$ -torka  $(e_1, e_2, e_3, \dots, e_n)$ , naziva se vezom.

#### Uloga (*role*)

funkcija koju skup entiteta obavlja u skupu veza.

### Skup vrijednosti, atribut

- Atribut je funkcija koja preslikava iz skupa entiteta ili skupa veza u skup vrijednosti ili Kartezijev produkt skupova vrijednosti:

$$f: E_i \rightarrow V_i$$

$$f: E_i \rightarrow V_{i1} \times V_{i2} \times \dots \times V_{in}$$

$$f: R_i \rightarrow V_i$$

$$f: R_i \rightarrow V_{i1} \times V_{i2} \times \dots \times V_{in}$$

### Regularni i slabi entiteti

- regularni entitet je entitet koji može postojati sam za sebe
- slabi entiteti (engl. *weak entity*) ne postoje ukoliko ne postoji i neki drugi entitet (entitet vlasnik)
- slabi entiteti, osim što su egzistencijalno slabi, također mogu biti i identifikacijski slabi
  - kod određivanja identifikatora nisu im dovoljni vlastiti atributi
  - za identifikaciju se koriste i ključni atributi entiteta vlasnika

### Definicija 1. (Teorey)

U vezi koja povezuje entitete  $E_1, \dots, E_k, \dots, E_m$ , spojnost =1 entiteta  $E_k$  znači da za svaku vrijednost svih entiteta  $E_1, \dots, E_m$ , osim  $E_k$ , uvijek postoji najviše jedna vrijednost od  $E_k$ .

$\Rightarrow$  može se reći da tada vrijedi funkcijska zavisnost:

$$m \cup K_j \setminus K_k \rightarrow K_k$$

$$j = 1$$

gdje su skupovi  $K_j$ , ( $j = 1, \dots, m$ ) ključevi entiteta  $E_1, \dots, E_m$

### Vlastiti atributi entiteta

- vlastiti atribut entiteta je atribut koji opisuje znanja o entitetu koja se pripisuju isključivo samom entitetu, a nikako vezi s drugim entitetima
- isključivo identifikacijski slabi entiteti, osim svojih vlastitih atributa, posjeduju i attribute primarnog ključa entiteta vlasnika

### Protumjere

sigurnost baze podataka se osigurava zaštitom na nekoliko razina

#### • zaštita na razini SUBP

spriječiti pristup bazama podataka ili onim dijelovima baza podataka za koje korisnici nisu ovlašteni

#### • zaštita na razini operacijskog sustava

spriječiti pristup radnoj memoriji računala ili datotekama u kojima SUBP pohranjuje podatke

#### • zaštita na razini računalne mreže

spriječiti presretanje poruka (*sniffing*) na internetu i intranetu

#### • fizička zaštita

fizički zaštititi lokaciju računalnog sustava

#### • zaštita na razini korisnika

spriječiti da ovlašteni korisnici nepažnjom ili namjerno (npr. U zamjenu za mito ili druge usluge) omoguće pristup podacima neovlaštenim osobama

### Vrste dozvola u SQL-u na razini baze podataka(*dbPrivilege*)

#### • CONNECT

uspostavljanje SQL-sjednice i obavljanje operacija nad objektima za koje je korisnik dobio dozvolu od vlasnika objekta ili je njihov vlasnik, kreiranje virtualnih i privremenih relacija

#### • RESOURCE

CONNECT + kreiranje novih relacija u bazi podataka

#### • DBA

RESOURCE + neovisno o vlasništvu i dozvolama nad objektima u bazi podataka: sve vrste operacija nad svim objektima, uništavanje svih objekata (uključujući i bazu podataka)

- korisnik koji kreira bazu podataka je vlasnik te baze podataka i implicitno dobiva DBA (*Database administrator*) dozvolu

### Vrste dozvola u SQL-u na razini [virtualne] relacije(*tablePrivilege*)

SELECT [(*columnList*)], UPDATE [(*columnList*)], INSERT, DELETE, REFERENCES [(*columnList*)], INDEXS, ALTER, ALL PRIVILEGES

### Pohranjene procedure (pohranjene funkcije)

- Pohranjena procedura ili pohranjena funkcija je potprogram koji je pohranjen u rječniku podataka i koji se izvršava u kontekstu sustava za upravljanje bazama podataka
  - može se promatrati kao procedura ili funkcija kojom se proširuje skup SQL funkcija ugrađenih u SUBP
- procedura je potprogram koji u pozivajući program ne vraća rezultat
- funkcija je potprogram koji u pozivajući program vraća rezultat

### Prednosti uporabe pohranjenih procedura

- proširenje mogućnosti SQL jezika
- omogućena je zaštita podataka na razini funkcije (a ne samo objekta)
- omogućena je uporaba klijent-poslužitelj arhitekture oslonjene na poslužitelj:
  - postiže se veća učinkovitost SUBP
- SUBP ne mora ponavljati prevođenje i optimiranje SQL upita
  - postiže se veća produktivnost programera i smanjuje mogućnost pogreške
- programski kôd potreban za obavljanje nekog postupka koji čini logičku cjelinu implementira se i testira na samo jednom mjestu

Nabrojite zadaće sustava za upravljanje bazama podataka

- sakriva od korisnika detalje fizičke pohrane podataka
- omogućuje definiciju i rukovanje s podacima
  - DDL - Data Definition Language
  - DML - Data Manipulation Language
- obavlja funkciju zaštite podataka
  - integritet podataka
  - pristup podacima - autorizacija, sigurnost

- kontrola paralelnog pristupa
- obnova u slučaju razrušenja
- optimiranje upita