

Integritets- vilkor/regler



Integritet (Integrity)

Finns tre typer av villkor: Inneboende, Implicita och Explicita

- Är primärnyckeln alltid unik?
- Har främmande nyckeln alltid en referens?
- Två tabeller, ett för konton och ett för lösenorden. Vad händer med lösenorden om konto tas bort?
- Databashanteraren underlättar jobbet!

Vanliga exempel

```
CREATE TABLE cars (cars_id int(11) PRIMARY KEY, owner int(11) NOT NULL);
```

```
CREATE TABLE employee (employee_id int(11) PRIMARY KEY NOT NULL, car  
int(11), FOREIGN KEY (car) REFERENCES cars(owner) ON DELETE SET NULL);
```

- NOT NULL garanterar inga null värden
- PRIMARY KEY garanterar endast unika värden
- FOREIGN KEY garanterar att det finns en bil innan vi lägger till den hos en anställd
- ON DELETE SET NULL sätter den anställdas bil nummer till null om bilen tas bort ur registret

Finns även..

- ON DELETE CASCADE
- ON DELETE SET DEFAULT
- ON UPDATE CASCADE

Läs om dessa i kursboken, sidan. 242

Kan skriva egna “vilkor”

- Databashanterare med triggers kallas aktiva databaser

```
CREATE TRIGGER save_old_user AFTER DELETE ON user
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO old_user VALUES (OLD.username);
```

```
END
```

Stored procedure

- För-kompilerad kod, dvs. snabbare körning
- Ökar säkerhet, en "procedure" kan ges till en användare (i databasen)
- Kan ta emot in-data
- Kan manipulera data utan att "se" tabellerna
- Till skillnad från vyer så är en stored procedure mycket mer än en data representation.

Exempel

```
CREATE PROCEDURE insert_user (IN username varchar(32)) BEGIN  
    IF (LENGTH(username) > 4) THEN  
        INSERT INTO user VALUES (username);  
    END IF;  
END  
  
CALL insert_user('Hello World User');
```