

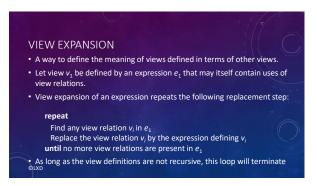




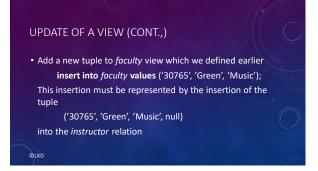




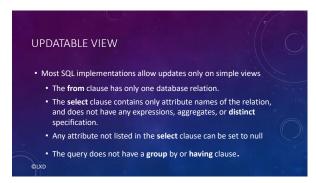
VIEWS DEFINED USING OTHER VIEWS One view may be used in the expression defining another view A view relation v₁ is said to depend directly on a view relation v₂ if v₂ is used in the expression defining v₁ A view relation v₁ is said to depend on view relation v₂ if either v₁ depends directly to v₂ or there is a path of dependencies from v₁ to v₂ A view relation v is said to be recursive if it depends on itself.











UPDATE OF A VIEW
- AND SOME NOT AT ALL

• create view history_instructors as select *
from instructor
where dept_name= 'History' ; WITH CHECK OPTION

• What happens if we insert ('25566', 'Brown', 'Biology', 100000) into history_instructors?

MATERIALIZED VIEWS物化视图

• Materializing a view

• create a physical table containing all the tuples in the result of the query defining the view

• If relations used in the query are updated, the materialized view result becomes out of date

• Need to maintain the view, by updating the view whenever the underlying relations are updated

• Materialized view maintenance 物化视图维护

OBJECTIVES

• Join Expressions

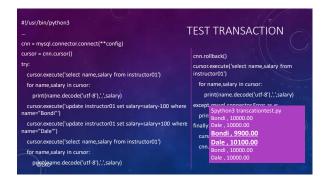
• Views

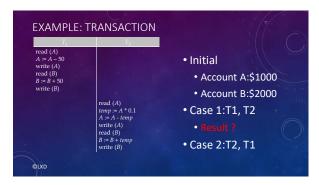
• Transactions

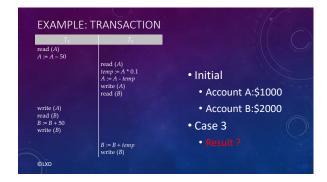
• Integrity Constraints

• SQL Data Types and Schemas

• Authorization









INTEGRITY CONSTRAINTS完整性约束

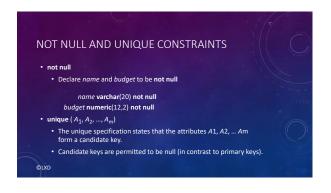
• Integrity constraints guard against accidental damage to the database, by ensuring that authorized changes to the database do not result in a loss of data consistency.

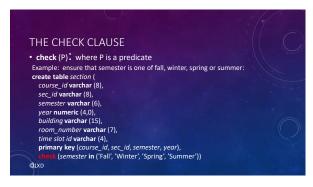
• A checking account must have a balance greater than \$10,000.00

• A salary of a bank employee must be at least \$4.00 an hour

• A customer must have a (non-null) phone number







EFERENTIAL INTEGRITY参照完整性

• Ensures that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

• Example: If "Biology" is a department name appearing in one of the tuples in the instructor relation, then there exists a tuple in the department relation for "Biology".

• Let A be a set of attributes. Let R and S be two relations that contain attributes A and where A is the primary key of S. A is said to be a tordital large of R if for any values of A appearing in R these values also appear in S.

INTEGRITY CONSTRAINT VIOLATION DURING
TRANSACTIONS

• E.g.
create table person (
ID char(10),
name char(40),
mother char(10),
fother char(10),
primary key ID,
foreign key father references person,
foreign key mother references person)

INTEGRITY CONSTRAINT VIOLATION DURING TRANSACTIONS (CONT.,)

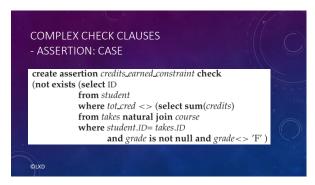
• How to insert a tuple without causing constraint violation?

• insert father and mother of a person before inserting person

• OR, set father and mother to null initially, update after inserting all persons (not possible if father and mother attributes declared to be not null)

• OR defer 延迟 constraint checking: Initially deferred







BUILT-IN DATA TYPES IN SQL

• date: Dates, containing a (4 digit) year, month and date

• Example: date '2005-7-27'

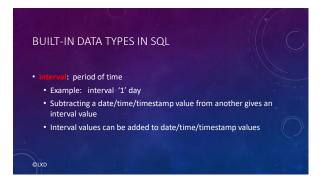
• time: Time of day, in hours, minutes and seconds.

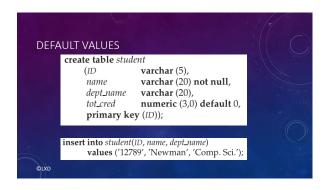
• Example: time '09:00:30' time '09:00:30.75'

• timestamp: date plus time of day

• Example: timestamp '2005-7-27 09:00:30.75'





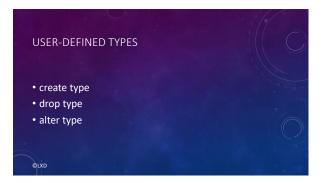






















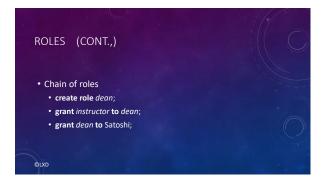






ROLES角色

• create role instructor;
• grant instructor to Amit;
• Privileges can be granted to roles:
• grant select on takes to instructor;
• Roles can be granted to users, as well as to other roles
• create role teaching_assistant
• grant teaching_assistant to instructor;
• Instructor inherits all privileges of teaching_assistant



AUTHORIZATION ON VIEWS
create view geo_instructor as
(select *
from instructor
where dept_name = 'Geology');
grant select on geo_instructor to geo_staff









