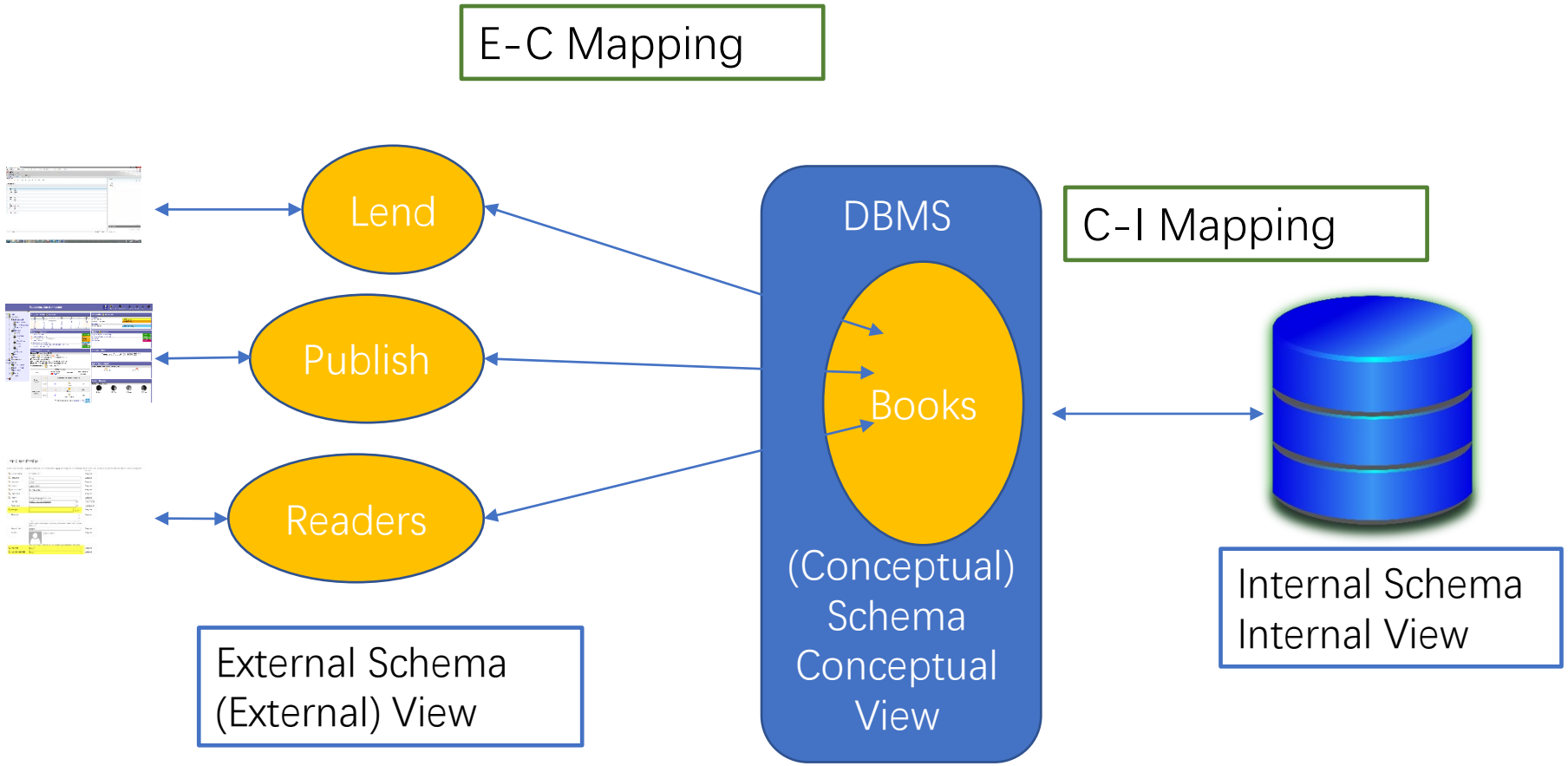
Lecture 6 SQL- View, Integrity

CS211 - Introduction to Database

View

CS211 – Introduction to Database

3-schema/view



- Logic data independence
- Physical data independence

view

- Data in a table is physically stored
- Tables ←→ files

- View is defined basing on conceptual schema
- Only definition (External-Conceptual mapping) is stored

- Data changing in table may cause view changing
- Will data changing in view cause table data change ? Depends

Create View

Instructor and the course they taught (Lecture)

iName, cName, credit

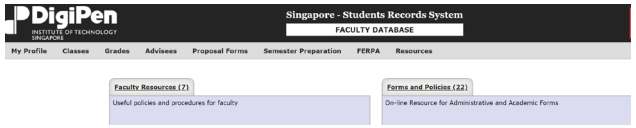
iID	iName	dID	workload
101	Jeremy	03	2.5
102	Vadim	03	2
103	Prabhu	03	1.5
104	Liu Fang	03	1.5

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

```
CREATE VIEW Lecture AS
    ( SELECT iName, cName, credit
      FROM instructor, course
      WHERE instructor.iID=course.iID);
```

Logic Data Independence

SRS



exec “select cName from lecture;”

Lecture(
iName
cName
credit)

External view

E-C Mapping:

```
CREATE VIEW Lecture AS
( SELECT iName,
         cName,
         credit
  FROM instructor,
         course
  WHERE
       instructor.iID=
       course.iID )   ;
```

instructor(
iID
iName
dID
workload)

course(
cID
cName
hours
…)

…

Conceptual view

If course.cname is changed to course.name
mapping should also be changed: create view … (select iName, name as cName …).

Schema of view remain unchanged, so that the statement in application “select cName from lecture” could also remain unchanged.

Create View

Students from CS department

(compStu)

schema

```
dept(
  dID
  dName
  dean)
```

```
CREATE VIEW compStu AS
    ( SELECT *
      FROM student
      WHERE student.dID IN (SELECT dept.dID
                            FROM dept
                            WHERE dept.dName='CS');
```

Query View

Instructor and the course they taught (Lecture)
iName, cName, credit

```
SELECT *
FROM Lecture
WHERE credit > 3
```

Students from CS department and under 20 (compStu)

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

```
SELECT *
FROM compStu
WHERE age<20
```

View Query Implementation

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

Students from CS department and under 20 (compStu)

```
SELECT *  FROM compStu  WHERE age<20;
```

Substitute definition of view compStu into above expression

```
SELECT *  FROM ( SELECT * FROM Student WHERE student.dID IN
(SELECT dept.dID  FROM dept  WHERE dept.dName='CS')) R
) WHERE age<20;
```

Optimize

```
SELECT *  FROM Student, dept  WHERE dept.dID=student.dID
AND dept.dName='CS' AND age<20
```

Create View – aggregation func

statStud(sID, sName, avgS, minS, maxS, cnt)

```
CREATE VIEW statStud AS
    ( SELECT s.sID, s.sName, avg(score) avgS, min(score) minS,
                max(score) maxS, count(*) cnt
      FROM Student s, rc
      WHERE s.sID = rc.sID
      GROUP BY s.SID );
```

sID	cID	score
4001	211	87.5
4001	225	94
4001	228	78
4002	211	92
4003	225	65
4003	228	74
4004	211	88
4004	225	82.4
4005	225	86.7
4005	228	89

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

Update View

statStud(sID, sName, avgS, minS, MaxS, cnt)

UPDATE statStud SET avgS=85 WHERE sID= '4001';

Insert into View

```
CREATE VIEW depStud(sName,dID) AS
  (SELECT sName, dID FROM Student);
```

```
INSERT INTO depStud VALUES ('Tang', '03')
```

Accept: When sID is not specified as a primary key

Reject: if sID is the primary key. (Primary key sID in student can not be null, according to Entity Integrity)

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

View Modification – not allowed

When creating view, use

- Select aggregation function
- Select distinct
- group by
- Select arithmetic expression
- All columns are from a single table but without Primary Key

View Modification

CREATE VIEW Cstud(sID, sName, grade)

AS (SELECT sID, sName, grade FROM Student WHERE dID= '03')

INSERT INTO Cstud VALUES ('4008', 'Tang', 'Freshman')

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

Delete a view

DROP VIEW view

Drop external schema and the E-C Mapping
Without removing data from the table

Integrity

CS211 – Introduction to Database

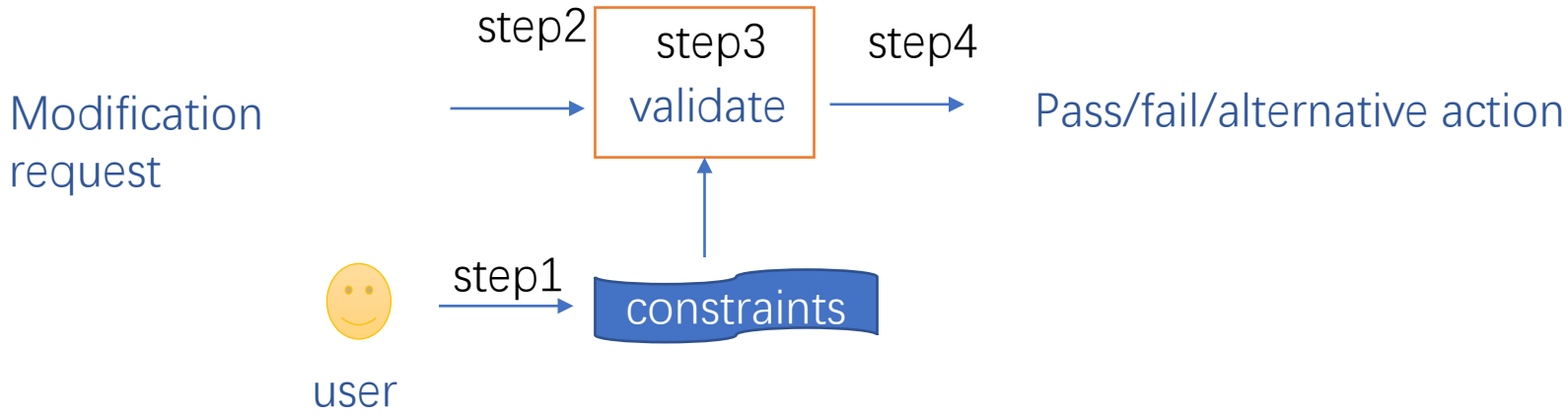
Database Integrity

- **Data integrity** is the maintenance of, and the assurance of the accuracy and consistency of, data over its entire life-cycle

- What causes the violation of data integrity
  - Wrong input
  - Wrong programming operations
  - …

Keep Integrity

- DBMS: Automatically keep integrity
  - Step1: User define integrity constraint (DDL)
  - Step2: User issue update request
  - Step3: Integrity Control - DBMS validate the request basing on the defined constraint
  - Setp4: DBMS pass/fail the request



Integrity constraint

- Four Components
  - Object: data collection
  - Predicate condition: constraint
  - Trigger: when to validate
  - Response: action when fail the validation

Integrity Constraint

- Object
  - Attribute level: age
  - Table level: hour/credit
- Predicate condition
  - Age>15 and age<=40
  - (hour/credit)>=6 and (hour/credit)<=7
- Trigger
  - Update/Insert into table
- Response
  - Reject the request of modification

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

age>15 and age<=40

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

(hour/credit)>=6 and
(hour/credit)<=7

Integrity – static vs. dynamic

Static: fixed value, fixed range

age > 0 and age < 150

Dynamic

Salary, only allow to be increased

e.g. salary could be 6000, or 8000

change 6000 → 8000 pass!

change 8000 → 6000 reject!

Integrity Constraint Definition

- Method 1: CONSTRAITNT
  - When define a static constraint
  - When the object is an attributes in a table

  - Entity integrity - Primary key
  - Referential integrity - Foreign key
  - User defined integrity

- Method 2: TRIGGER
  - When define a dynamic constraint
  - When the object involves more than one table

SQL Constraint – Primary key

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

- CREATE TABLE

```
CREATE TABLE student(
  sID char(4) PRIMARY KEY,
  sName char(10),
  gender char(1),
  age int,
  dID char(2),
  grade char(10)
)
```

```
CREATE TABLE student(
  sID char(4),
  sName char(10),
  gender char(1),
  age int,
  dID char(2),
  grade char(10),
  [CONSTRAINT pk_stu] PRIMARY KEY(sID) );
```

SQL Constraint – Add primary key

- ALTER TABLE

```
CREATE TABLE student(
  sID char(4),
  sName char(10),
  gender char(1),
  age int,
  dID char(2),
  grade char(10) )
```

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

```
ALTER TABLE student
ADD [CONSTRAIT pk_stu]
PRIMARY KEY (sID);
```

SQL Constraint – Drop primary key

- ALTER TABLE

MariaDB/MySQL

```
ALTER TABLE student DROP PRIMARY KEY;
```

SQLServer/Oracle

```
ALTER TABLE student DROP CONSTRAINT pk_stu;
```

sID	sName	gender	age	dID	grade
4001	Amy	F	20	03	Sophomore
4002	Alice	F	19	04	Freshman
4003	Bob	M	20	03	Sophomore
4004	Cathy	F	18	04	Freshman
4005	John	M	21	03	Junior

SQL Constraint – composite primary key

```
CREATE TABLE rc(
  sID char(4),
  cID char(3),
  score float,
 [CONSTRAINT pk_stu] PRIMARY KEY (sID, cID) );
```

sID	cID	score
4001	211	87.5
4001	225	94
4001	228	78
4002	211	92
4003	225	65
4003	228	74
4004	211	88
4004	225	82.4
4005	225	86.7
4005	228	89

SQL Constraint – foreign key

Student.sid, course.cid
Must be primary key

```
CREATE TABLE rc(
  sID char(4),
  cID char(3),
  score float,
  PRIMARY KEY (sID, cID),
  FOREIGN KEY (sID) REFERENCES student(sID),
  FOREIGN KEY (cID) REFERENCES course(cID) ON DELETE CASCADE)
```

sID	cID	score
4001	211	87.5
4001	225	94
4001	228	78
4002	211	92
4003	225	65
4003	228	74
4004	211	88
4004	225	82.4
4005	225	86.7
4005	228	89

SQL Constraint – foreign key - cascade

FOREIGN KEY (cID) REFERENCES course(cID) ON DELETE CASCADE

DELETE FROM course WHERE cid='211';

All tuples with cid='211' in rc will be deleted automatically

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

sID	cID	score
4001	211	87.5
4001	225	94
4001	228	78
4002	211	92
4003	225	65
4003	228	74
4004	211	88
4004	225	82.4
4005	225	86.7
4005	228	89

SQL Constraint – User defined integrity

```
dept(
  dID
  dName
  dean)
```

```
CREATE TABLE student(
  sID char(4) PRIMARY KEY,     sName char(10),
  gender char(1) CHECK(gender= '1' or gender = '0'),
  age int CHECK(age>=1 and age<150),
  dID char(2),
  grade char(10),
  FOREIGN KEY (dID) REFERENCES dept(dID) ON DELETE CASCADE )
```

SQL Constraint – User defined integrity

If table student exists already

```
ALTER TABLE student ADD CONSTRAINT
CHECK(age>=15 and age<=150);
```

SQL Constraint - User defined integrity

- Course – credit [0..5], iID F.K. with on delete cascade, references to instructor.iid

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

```
CREATE TABLE course(
  cID char(3),
  cName char(12),
  hours int,
  credit float
  iID char(3),
                                                    ,
                                                    ) ;
```

SQL Constraint - User defined integrity

- Course – credit [0..5], iID F.K. with on delete cascade, references to instructor.iid

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

```
CREATE TABLE course(
  cID char(3),
  cName char(12),
  hours int,
  credit float  CHECK(credit>=0 and credit<=5),
  iID char(3),
  FOREIGN KEY (iID) FREFERENCES instructor(iID) ON DELETE CASCADE);
```

SQL Constraint - User defined integrity

- Course – multiple attributes constraint

cID	cName	hours	credit	iID
211	Database	56	3	104
225	C++	56	4	102
228	OS	56	3	103

```
CREATE TABLE course(
  cID char(3) PRIMARY KEY,
  cName char(12),
  hours int,
  credit float CHECK(credit>=0 and credit<=5),
  iID int(11),
  CONSTRAINT ctcc CHECK(hours/credit > 10),
  FOREIGN KEY (iID) REFERENCES instructor(iID) ON DELETE CASCADE
)
```

Resource

- MairaDB, Create Table
  - https://mariadb.com/kb/en/create-table/

- Constraint Expression in Create Table
  - https://mariadb.com/kb/en/create-table/#constraint-expressions