Basic Relational Data Model
Keys & Constraints

Hafiz Tayyeb Javed

Assistant Professor (CS)

Department of Computer Science

FAST-NU, CFD Campus

The Referential Integrity Constraint

A **referential integrity constraint** is a statement that limits the values of the foreign key to those already existing as primary key values in the corresponding relation

Foreign Key with a Referential Integrity Constraint

NOTE: The primary key of the relation is underlined and any foreign keys are in *italics* in the relations below:

ITEM (Item#, Item_Name, Department, Buyer)
ORDER_ITEM (OrderNumber, *Item#*, Quantity, Price, ExtendedPrice)

Where ORDER_ITEM.ITEM# must exist in ITEM.ITEM#
OrderNumber is an alternate name of InvoiceNumber

Integrity *(Fig. from C.J. Date)*

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Master/ Parent – Child Tables

Master Tables
DEPARTMENT (**DeptID**, DeptName, BudgetCode, ManagerName, Loc, TelNo)
CUSTOMER(**CustId**, Name, Address, Telno, Email)
Part (**Part#**, PName, Weight, MakeYear, Price, PType, RefPart#)
Product (**Item#**, Name, Price, Department, Buyer)
Course(**CourseID**, Name, CrHrs)
Semester(**SemID**, StartDate, EndDate, SGPA)
Student(**StuId**, FName, LName, DOB, NIC#, Address, City, Telno, Deptno)
Patient(**PatID**, FName, LName, DOB, Address, City, Telno, NIC#, PType)
ATM(**Machine#**, MName, MType, Size, Brand, Colour)

Some of the above table's and their Child Tables (FK with and without Comp. Keys)
CourseReg(**StuID**, **SemID**, **CourseID**, Sec#, Grade)
PatientVisit(**PatID**, **DependSNO**, **VisitSNO**, VDate, DocID, Diagnosis, VStatus)
EMPLOYEE(**EmpNo**, EmpName, DOB, Addr, City, TelNo, Job, HireDate, *DeptID*)
ATMTransaction (**Card#, Serial#,** Amount, TransDate, TransType, Machine#)

Integrity Rules

Entity integrity

No two rows with the same primary key value
No null values in a primary key

Referential integrity

Foreign keys must match candidate key of source table
Foreign keys in some cases can be null
The database must not contain any unmatched foreign key values.
If B references A, A must exist.

Referential Actions (Cont.)

When we update a tuple from a table, say S, that is *referenced* by another table, say SP,
There are similar choices of *referential actions*:

ON UPDATE CASCADE
ON UPDATE RESTRICT

There could be other choices besides these three. , e.g.,

ON UPDATE SET DEFAULT.
ON UPDATE SET NULL.

Composite Keys

Combination of primary keys of selected attributes gives concept of composite key. composite key is a extended form of primary key

Example: For one semester

{SID, CID }

COURSE-ALLOCATION

SID	CID	SEC#	INST_ID	B#	ROOM#

Alternatively, for all semesters

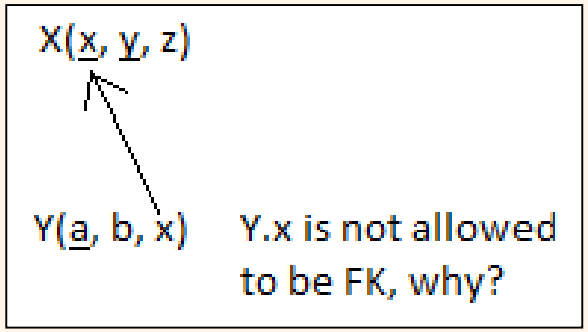
SEM-ALLOCAT(SEMID, CID, SEC#, BUILDING#, ROOM#,INST_ID)

Following is not valid because duplicate rows and blank columns can be possible

SEM-ALLOCAT(ID, SEMID, CID, SEC#, BUILDING#, ROOM#,INST_ID)

CID data value will be like ISE361, ICS334 etc

Composite Keys …



Course(CourseID, Name, CrHrs)
Semester(SemID, StartDate, EndDate)
Student(StuId, FName, LName, DOB, …)
…

Courses are issued or allocated for registration

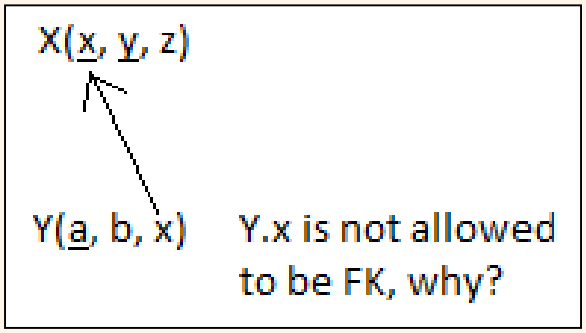
CourseIssue(SemID, CID, Sec#, InstID, B#, R#, Days, Time, MaxStu)

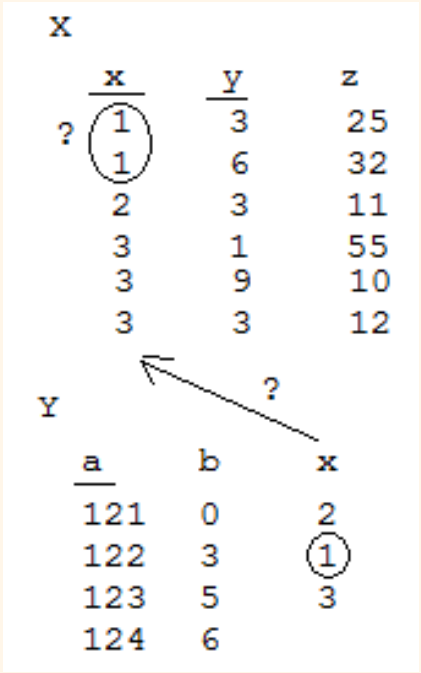
Students will register in issued courses:

CourseReg(StuID, SemID, CID, Sec#, Grade)
(Student can see instructor of course by using SQL, SemID, CID and Sec# will be FK refer to CourseIssue)

Meaningful Learning

Analysis and Semantics Evaluations





Semantics: Creating two tables with relevant data
Analysis: Y.x=1 cannot be mapped exactly with X.x records 1, 3 or 1, 6

Evaluations: Default FK Y.x cannot be created. Trigger can be created on Y.x to make sure Y.x will be a subset of X.x. Otherwise, data can be put in Y.x which does not belong to or mapped to X.x.

Composite Key…

Do NOT make composite Key

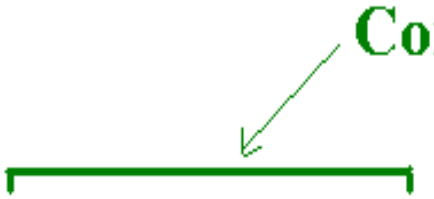
Department

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

PK

Meaningful Learning?

Composite Key ...



OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300	300
2000	101100	4	50	200
3000	101100	2	50	100
2000	101200	2	50	100
3000	101200	1	50	50
1000	201000	1	300	300
1000	202000	1	130	130

```
alter table ORDER_ITEM
add constraints pk_ordItem PRIMARY KEY(OrderNumber, SKU);
```

How data is stored in a table?

Invoice#: 3000
Date: 23-MAR-2010

Customer:
Ejaz Ahmed (#2578412)
25, Main Street, Islamabad
Pakistan, Tel. 051-230-5874

SKU	Desc	Unit Price	Qty	Extend. Price
100200	Std. Scuba Tank, Yellow	300.00	1	300.00
101100	Std. Scuba Tank, Magenta	50.00	2	100.00
101200	Dive Mask, Med Clear	50.00	1	50.00
	Total:			450.00

How data is shown to a user?

Composite Keys and Foreign Keys

PatientVisit(PatID, VisitSNO, Vdate, DocID, Diagnosis,…)

LabTest(*PatID, VisitSNO*, TestID, Tdate, …)

LabTest has FKs ***PatID, VisitSNO***, referring to same corresponding composite keys in PatientVisit

Indexing Columns – Comp. Keys

PATIENT_VISIT Without Indexing

PATIENT_ID	DEPEND_SN
0222082	0
0257421	0
1840190	6
1976193	0
6901319	0
6900626	0
6911102	0
0213919	0
1731899	0
6781521	8
RS00079	2
0207842	0
6781521	0
1770371	0
0227638	0
1926081	1
6751368	2

PATIENT_VISIT Keys/ Indexed

PATIENT_ID	DEPEND_SN	PATIENT_VIS	EMP_ID	DOCTOR_SN	VISIT_DATE	VISIT_N
0204531	0	15	6772356	28	10/3/2007 12:15:51 PM	
0204531	0	16	7073498	19	4/29/2008 1:40:11 PM	
0204531	0	17	6772356	37	4/29/2008 2:09:45 PM	
0204531	0	18	7024037	16	5/31/2008 9:20:18 AM	
0204531	0	19	7024037	4	8/5/2008 1:00:34 PM	
0204531	0	20	6772356	15	8/9/2008 8:36:01 AM	
0204531	0	21	7024037	4	8/11/2008 8:38:44 AM	
0204531	0	22	7073498	16	8/25/2008 1:01:39 PM	
0204531	0	23	6772356	7	10/26/2008 2:32:56 PM	
0204531	0	24	6772356	7	10/28/2008 8:56:20 AM	
0204531	0	25	7024037	39	11/10/2008 1:27:23 PM	
0204532	0	1	6850269	30	9/11/2005 7:17:21 PM	
0204532	0	3	7024037	50	10/5/2005 2:13:42 PM	
0204532	0	4	6850269	50	12/14/2005 2:46:16 PM	
0204532	0	5	6850269	59	1/24/2006 9:41:49 PM	
0204532	1	1	6850269	41	8/6/2005 2:55:29 PM	
0204532	1	2	7043508	31	9/6/2005 2:59:43 PM	
0204532	1	3	7043508	27	9/7/2005 3:25:09 PM	
0204532	1	4	6900551	12	9/8/2005 11:19:55 AM	
0204532	1	6	6971354	12	9/12/2005 11:12:52 AM	
0204532	1	7	6850269	57	1/24/2006 9:27:27 PM	
0204532	1	8	7024037	26	2/7/2006 2:56:09 PM	
0204532	2	1	7033244	3	10/22/2005 3:34:27 PM	
0204532	2	2	6971354	44	1/18/2006 2:54:51 PM	
0204573	0	1	7024037	35	1/21/2006 11:02:11 AM	

Other Constraints

Null: Represents a value of an attribute that is currently unknown or is no applicable for this tuple, means nothing.

Entity integrity Constraint: In a base relation, no attribute of a primary key can be ***null***.

Referential Integrity Constraint: If a foreign key exists in a relation, either the foreign key value ***must match a candidate key*** (or PK) value of some tuple in its home relation or the foreign key value must be ***wholly null***.

Domain Constraint: Specifies that the value of attribute A must be an atomic value from the domain ***DOM(A).***

Additional Constraints: Default, NOT NULL, UNIQUE, CHECK; Business Constraints using DB triggers and client's programming. For performance Indexes are also constraints.

Type or Domain Constraints (Cont.)

The **check** clause in SQL permits domains to be restricted:

Use **check** clause to ensure that an hourly-wage domain allows only values greater than a specified value.

create domain *hourly-wage* **numeric(5,2)**
constraint *value-test* **check**(*value* > = 4.00)

The domain *hourly-wage* is declared to be a decimal number with 5 digits, 2 of which are after the decimal point

The domain has a constraint that ensures that the hourly-wage is greater than 4.00

Attribute Constraints

An attribute constraint is just a declaration to the effect that a specified attribute is of a specified type.
For example:

create table *account*
(branch-name char(15),
account-number char(10) **not null,**
balance integer,
……)

Attribute constraints are part of the definition of the attribute.
Any attempt to introduce an attribute value into the database that is not a type of the relevant type will simply rejected.

Such a situation should never arise.

Table Constraints

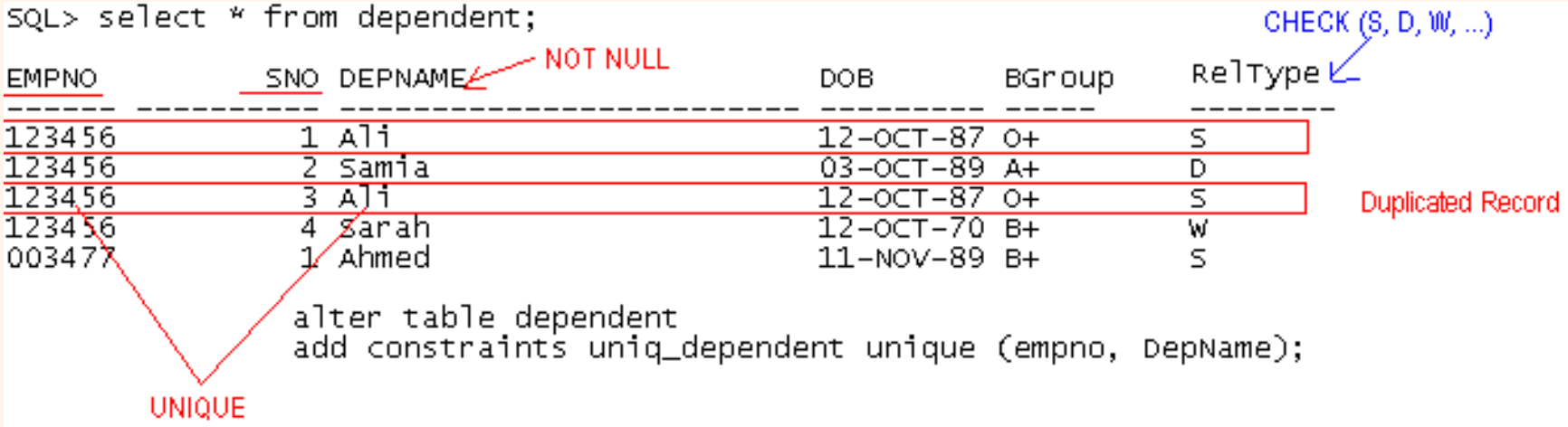
A table constraint is a constraint on an individual table.

Example:

Suppliers in London must have status 20.
Two attributes, CITY and STATUS, of table S are involved.

Other Constraints Examples

Unique constraint can be applied on single or multiple columns



CHECK (Max_Participants>=Enrolled_Participants)

CHECK (Deptno>0)

CHECK (RelType IN ('S', 'D', 'W'))

Default 'Y'

Database Triggers