Nulls and 3-valued logic

- **Conditions**: $x \ op \ y$ (where $op$ is $<$, $>$, $<=$, $=$, etc.) has value unknown $(U)$ when either $x$ or $y$ is null
  - WHERE $T.\text{cost} > T.\text{price}$
- **Arithmetic expression**: $x \ op \ y$ (where $op$ is $+$, $-$, $*$, etc.) has value NULL if $x$ or $y$ is NULL
  - WHERE $(T.\text{price}/T.\text{cost}) > 2$
- **Aggregates**: COUNT counts NULLs like any other value; other aggregates ignore NULLs

```sql
SELECT COUNT(T.CrsCode), AVG(T.Grade)
FROM Transcript T
WHERE T.StudId = '1234'
```
Nulls (cont’d)

- WHERE clause uses a three-valued logic – T, F, U(defined) – to filter rows. Portion of truth table:

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C1 AND C2</th>
<th>C1 OR C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>U</td>
<td>U</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>U</td>
<td>F</td>
<td>U</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

- Rows are discarded if WHERE condition is F(alse) or U(nknown)
  - e.g., WHERE T.CrsCode = ‘CS305’ AND T.Grade > 2.5
  - If a CHECK clause evaluates to unknown, the integrity constraints are considered to be observed.

Modifying Tables – Insert

- Inserting a single row into a table
  - Attribute list can be omitted if it is the same as in CREATE TABLE (but do not omit it)
  - NULL and DEFAULT values can be specified

```sql
INSERT INTO Transcript(StudId, CrsCode, Semester, Grade)
VALUES (12345, ‘CSE305’, ‘S2000’, NULL)
```
Bulk Insertion

• Insert the rows output by a SELECT

CREATE TABLE DeansList (  
    StudId     INTEGER,  
    Credits    INTEGER,  
    CumGpa     FLOAT,  
    PRIMARY KEY StudId  
)

INSERT INTO DeansList (StudId, Credits, CumGpa)  
SELECT T.StudId, 3 * COUNT(*), AVG(T.Grade)  
FROM Transcript T  
GROUP BY T.StudId  
HAVING AVG(T.Grade) > 3.5 AND COUNT(*) > 30

Modifying Tables – Delete and Update

• DELETE is similar to SELECT except:
  • No project list in DELETE clause
  • No Cartesian product in FROM clause (only 1 table name)
  • Rows satisfying WHERE clause (general form, including subqueries, allowed) are deleted instead of output

DELETE FROM Transcript T  
WHERE T.Grade IS NULL AND T.Semester <> 'S2000'

• Updates rows in a single table
  • All rows satisfying WHERE clause (general form, including subqueries, allowed) are updated

UPDATE Employee E  
SET E.Salary = E.Salary * 1.05  
WHERE E.Department = 'R&D'
Updating Views

- **Question**: Since views look like tables to users, can they be updated?
- **Answer**: Yes – a view update changes the underlying base table to produce the requested change to the view.

```
CREATE VIEW CsReg (StudId, CrsCode, Semester) AS
SELECT T.StudId, T.CrsCode, T.Semester
FROM Transcript T
WHERE T.CrsCode LIKE 'CS%' AND T.Semester='S2000'
```

Updating Views - Problem 1

```
INSERT INTO CsReg (StudId, CrsCode, Semester)
VALUES (1111, 'CSE305', 'S2000')
```

- **Question**: What value should be placed in attributes of underlying table that have been projected out (e.g., Grade)?
- **Answer**: Simple. NULL (assuming null allowed in the missing attribute) or DEFAULT.
Updating Views - Problem 2

- **Problem**: New tuple not in view
- **Solution**: Allow insertion (assuming the WITH CHECK OPTION clause has not been appended to the CREATE VIEW statement)

```sql
INSERT INTO CsReg (StudId, CrsCode, Semester)
VALUES (1111, 'ECO105', 'S2000')
```

This is not CS course!!

Updating Views - Problem 3

- Update to a view might *not uniquely* specify the change to the base table(s) that results in the desired modification of the view (ambiguity)

```sql
CREATE VIEW ProfDept (PrName, DeName)
AS
SELECT P.Name, D.Name
FROM Professor P, Department D
WHERE P.DeptId = D.DeptId
```

- Tuple <Smith, CS> can be deleted from ProfDept by:
  - Deleting row for Smith from Professor (but this is inappropriate if he is still at the University)
  - Deleting row for CS from Department (not what is intended)
  - Updating row for Smith in Professor by setting DeptId to null (seems like a good idea, but how would the computer know?)
Updating Views - Problem 3 (Cont’d)

- What to do at deleting <101202303, 123454321>>
  - Delete two rows from Teaching?
  - Delete two rows from Transcript?
  - Delete one with CS315 from Teaching and one with CS305 Transcript from Transcript? Or, CS305 from Teaching and CS315 from Transcript?

Updatable Views - Restrictions

- Updatable views are restricted to those in which
  - No Cartesian product in FROM clause
  - No aggregates, GROUP BY, HAVING, set operation
  - No expressions, DISTINCT keyword in the SELECT clause
  - ...

For example, if we allowed:

```
CREATE VIEW AvgSalary (DeptId, Avg_Sal) AS
SELECT E.DeptId, AVG(E.Salary)
FROM Employee E
GROUP BY E.DeptId
```

then how do we handle:

```
UPDATE AvgSalary
SET Avg_Sal = 1.1 * Avg_Sal
```