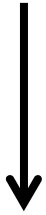
CHECK constraint

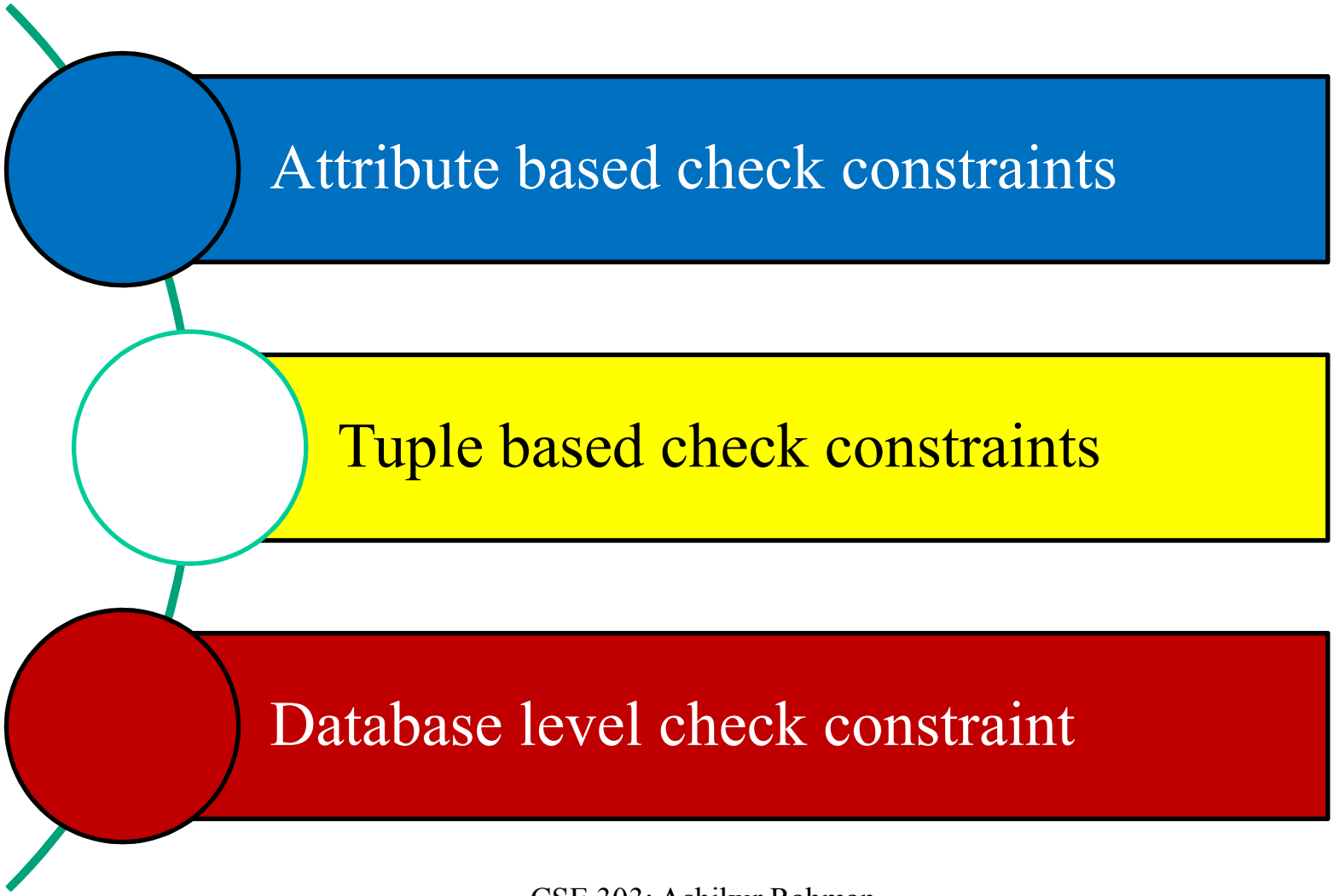
CHECK (<condition>)



CHECK (condition that must hold)

conditions are like predicates that are written after WHERE clause

CHECK constraint



Attribute based CHECK constraint

The speed of a laptop must be at least 2.0

```
CREATE TABLE Laptop(
        ….,
        speed NUMBER(3,2) CHECK(speed >= 2.0),
        ….
);
```

Attribute based CHECK constraint

The only types of printers are laser, ink-jet, and bubble-jet

```
CREATE TABLE Printer(
        ….,
        type CHAR(20) CHECK(type IN (‘laser’, ’ink-jet’, ’bubble-jet’)),
        ….
);
```

A model of a product must also be the model of a PC, a laptop, or a printer

Maker	Model	Type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
…	…	…
….	…	…

Product

MODEL	SPEED	RAM	HD	PRICE
1001	2.66	1024	250	2114
1002	2.1	512	250	995
1003	1.42	512	80	478
1004	2.8	1024	250	649
1005	3.2	512	250	630
…	…..	…	…	

PC

MODEL	SPEED	RAM	HD	SCREEN	PRICE
2001	**2**	**2048**	**240**	**20.1**	**3673**
2002	**1.73**	**1024**	**80**	**17**	**949**
2003	**1.8**	**512**	**60**	**15.4**	**549**
2004	**2**	**512**	**60**	**13.3**	**1150**
…	…..	….	….	…..	……

Laptop

Attribute based CHECK constraint

A model of a product must also be the model of a PC, a laptop, or a printer

```
CREATE TABLE Product(
        ….,
  model CHAR(4) CHECK(model IN (SELECT model FROM PC
                                UNION SELECT model FROM Laptop
                                 UNION SELECT model FROM Printer),
        ….
);
```

Tuple based CHECK constraint

If the Head of the Department's gender is male then his name must not begin with 'Ms.'

$P \rightarrow Q \iff \neg P \vee Q$

```
CREATE TABLE Department(
        ….,
        ….,
        PRIMAR KEY..
        FOREIGN KEY,…
        CHECK (gender = 'F' or name NOT LIKE 'Ms.%')
);
```

Tuple based CHECK constraint

A PC with a processor speed less than 2.0 must not sell for more than $800

$P \rightarrow Q \iff \neg P \vee Q$

```
CREATE TABLE PC(
        ….,
        ….,
        PRIMAR KEY..
        FOREIGN KEY,…
        CHECK (speed > 2.0 or price <= 800)
);
```

Tuple based CHECK constraint

A Laptop with a screen size less than 15 inches must have at least a 40 GB hard disk or sell for less than $1000

$P \rightarrow Q \iff \neg P \vee Q$

```
CREATE TABLE PC(
….,
      PRIMAR KEY..
      FOREIGN KEY,…
      CHECK (screen > 15 or hd > 40 or price < 1000)
);
```

If the relation Product mentions a model and its type then this model must appear in the relation appropriate to that type.

Maker	Model	Type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
…	…	…
….	…	…

Product

MODEL	SPEED	RAM	HD	PRICE
1001	2.66	1024	250	2114
1002	2.1	512	250	995
1003	1.42	512	80	478
1004	2.8	1024	250	649
1005	3.2	512	250	630
…	…..	…	…	

PC

MODEL	SPEED	RAM	HD	SCREEN	PRICE
2001	**2**	**2048**	**240**	**20.1**	**3673**
2002	**1.73**	**1024**	**80**	**17**	**949**
2003	**1.8**	**512**	**60**	**15.4**	**549**
2004	**2**	**512**	**60**	**13.3**	**1150**
….	….	…..	….	…..	…….

Laptop

Tuple based CHECK constraint

If the relation Product mentions a model and its type then this model must appear in the relation appropriate to that type.

```
CREATE TABLE Product(
….,
CHECK(

(type = ‘pc’ AND model IN (SELECT model FROM PC))
OR (type = ‘laptop’ AND model IN (SELECT model FROM Laptop))
OR (type = ‘printer’ AND model IN (SELECT model FROM Printer))

);
```

Database level CHECK constraint

- Database level CHECK constraints called assertions
- Condition checks over multiple relations
- Assertions are executed upon any modifications to any relations mentioned in the assertion
- SQL standard supports assertions

```
CREATE ASSERTION <assertion-name>

CHECK (<condition>);
```

```
CREATE ASSERTION <assertion-name>

CHECK (<condition>);
```

There are two approaches in writing conditions:

(1) Positive approach

(2) Negative approach

RULE: Departmental head must be an employee of the university

Department

name	head
CSE	2
EEE	4
ME	1

Employee

id	name	address	gender
1	A	X	M
2	B	Y	M
3	C	Z	F
4	D	W	M

Positive approach: Look at each departmental head and make sure that s/he is an employee of the university

Negative approach: Look for a departmental head who is not an employee and make sure that such head does not exist

EXISTS or NOT EXISTS which way to go?

Negative approach: Look for a departmental head who is not an employee and make sure that such head does not exist

Department

name	head
CSE	2
EEE	4
ME	1

Employee

id	name	address	gender
1	A	X	M
2	B	Y	M
3	C	Z	F
4	D	W	M

```
CREATE ASSERTION  checkHeadConstraint
CHECK (NOT EXISTS (SELECT head
                   FROM Department
                   WHERE head NOT IN (SELECT id
                                      FROM Employee)
);
```

Assertion more examples

No manufacturer of PCs may also make laptops

Maker	Model	Type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
…	…	…
….	…	…

Product

Assertion example

No manufacturer of PCs may also make laptops

'A manufacturer of a PC makes Laptop' such occurrences do NOT EXIST…

```
CREATE ASSERTION makeConstraint
CHECK (NOT EXISTS (SELECT maker
                   FROM Product
                   WHERE type = 'pc'
                    AND maker IN (SELECT maker
                                  FROM Product
                                 WHERE type = 'laptop')
);
```

If a laptop has a large main memory than a PC, then the laptop must also has a higher price than the PC

PC

MODEL	SPEED	RAM	HD	PRICE
1001	2.66	1024	250	2114
1002	2.1	512	250	995
1003	1.42	512	80	478
1004	2.8	1024	250	649
1005	3.2	512	250	630
…	…..	…	…	

Laptop

MODEL	SPEED	RAM	HD	SCREEN	PRICE
2001	**2**	**2048**	**240**	**20.1**	**3673**
2002	**1.73**	**1024**	**80**	**17**	**949**
2003	**1.8**	**512**	**60**	**15.4**	**549**
2004	**2**	**512**	**60**	**13.3**	**1150**
….	….	…..	….	…..	……

Assertion example

If a laptop has a large main memory than a PC, then the laptop must also has a higher price than the PC

```
CREATE ASSERTION makeConstraint
CHECK (NOT EXISTS (
                SELECT model
                FROM    Laptop, PC
                WHERE  Laptop.ram > PC.ram
                AND       Laptop.price < PC.price)
        )
);
```

If the relation Product mentions a model and its type then this model must appear in the relation appropriate to that type.

Maker	Model	Type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
…	…	…
….	…	…

Product

MODEL	SPEED	RAM	HD	PRICE
1001	2.66	1024	250	2114
1002	2.1	512	250	995
1003	1.42	512	80	478
1004	2.8	1024	250	649
1005	3.2	512	250	630
…	…..	…	…	

PC

MODEL	SPEED	RAM	HD	SCREEN	PRICE
2001	**2**	**2048**	**240**	**20.1**	**3673**
2002	**1.73**	**1024**	**80**	**17**	**949**
2003	**1.8**	**512**	**60**	**15.4**	**549**
2004	**2**	**512**	**60**	**13.3**	**1150**
….	….	…..	….	…..	……

Laptop

Assertion example

If the relation Product mentions a model and its type then this model must appear in the relation appropriate to that type.

```
CREATE ASSERTION makeConstraint
CHECK (NOT EXISTS (
            SELECT model
            FROM    Product
           WHERE  (type = 'pc' AND
                        model NOT IN (SELECT model FROM PC))
           OR (type = 'laptop' AND
                        model NOT IN (SELECT model FROM Laptop))
          OR (type = 'printer' AND
                        model NOT IN (SELECT model FROM Printer))
        )
);
```

A manufacturer of a PC also make a laptop with at least as great a processor speed

Maker	Model	Type
A	1001	pc
A	1002	pc
A	1003	pc
A	2004	laptop
A	2005	laptop
A	2006	laptop
B	1004	pc
B	1005	pc
B	1006	pc
B	2007	laptop
C	1007	pc
…	…	…
….	…	…

Product

MODEL	SPEED	RAM	HD	PRICE
1001	2.66	1024	250	2114
1002	2.1	512	250	995
1003	1.42	512	80	478
1004	2.8	1024	250	649
1005	3.2	512	250	630
…	…..	…	…	

PC

MODEL	SPEED	RAM	HD	SCREEN	PRICE
2001	**2**	**2048**	**240**	**20.1**	**3673**
2002	**1.73**	**1024**	**80**	**17**	**949**
2003	**1.8**	**512**	**60**	**15.4**	**549**
2004	**2**	**512**	**60**	**13.3**	**1150**
….	….	…..	….	…..	…….

Laptop

Assertion example

A manufacturer of a PC also make a laptop with at least as great a processor speed

```
CREATE ASSERTION makeConstraint
CHECK (NOT EXISTS (
                SELECT maker
                  FROM Product P1 NATURAL JOIN PC
                  WHERE speed >ALL (SELECT speed
                                    FROM Product P2 NATURAL JOIN Laptop
                                    WHERE P2.maker = P1.maker)
        )
);
```

Different Constraint Types

Type	Where Declared	When activated	Guaranteed to hold?
Attribute CHECK	with attribute	on insertion or update on attribute	not if subquery
Tuple CHECK	relation schema	insertion or update to relation	not if subquery
Assertion	database schema	on change to any relation mentioned	Yes

Giving Names to Constraints

Why give names? In order to be able to alter constraints.

Add the keyword CONSTRAINT and then a name:

(1) ssn CHAR(50) CONSTRAINT ssnIsKey PRIMARY KEY

(2) CONSTRAINT ninedigits CHECK (VALUE >= 100000000
AND VALUE <= 999999999)

(3) CONSTRAINT rightage CHECK (age >= 0 OR status = “dead”)

Altering Constraints

```
ALTER TABLE  Product  DROP CONSTRAINT positivePrice

ALTER TABLE  Product ADD CONSTRAINT
     positivePrice  CHECK  (price >= 0)


DROP ASSERTION  assert1
```