Creating and using a db system

(1) design schema

(2) create schema using DDL
- CREATE TABLE
- CREATE ASSERTION
- CREATE TRIGGER

(3) "Bulk load" initial data

(4) Repeat: execute queries and modifications on the stored data

Components of SQL

- schema definition
- data retrieval
- data modification
- indexes
- constraints
- views
- triggers
- transaction
- authorizations

Constraints in SQL:

- primary keys
- UNIQUE
- NOT NULL
- a referential-integrity constraint
- attribute-based constraints
- tuple-based constraints
- assertions

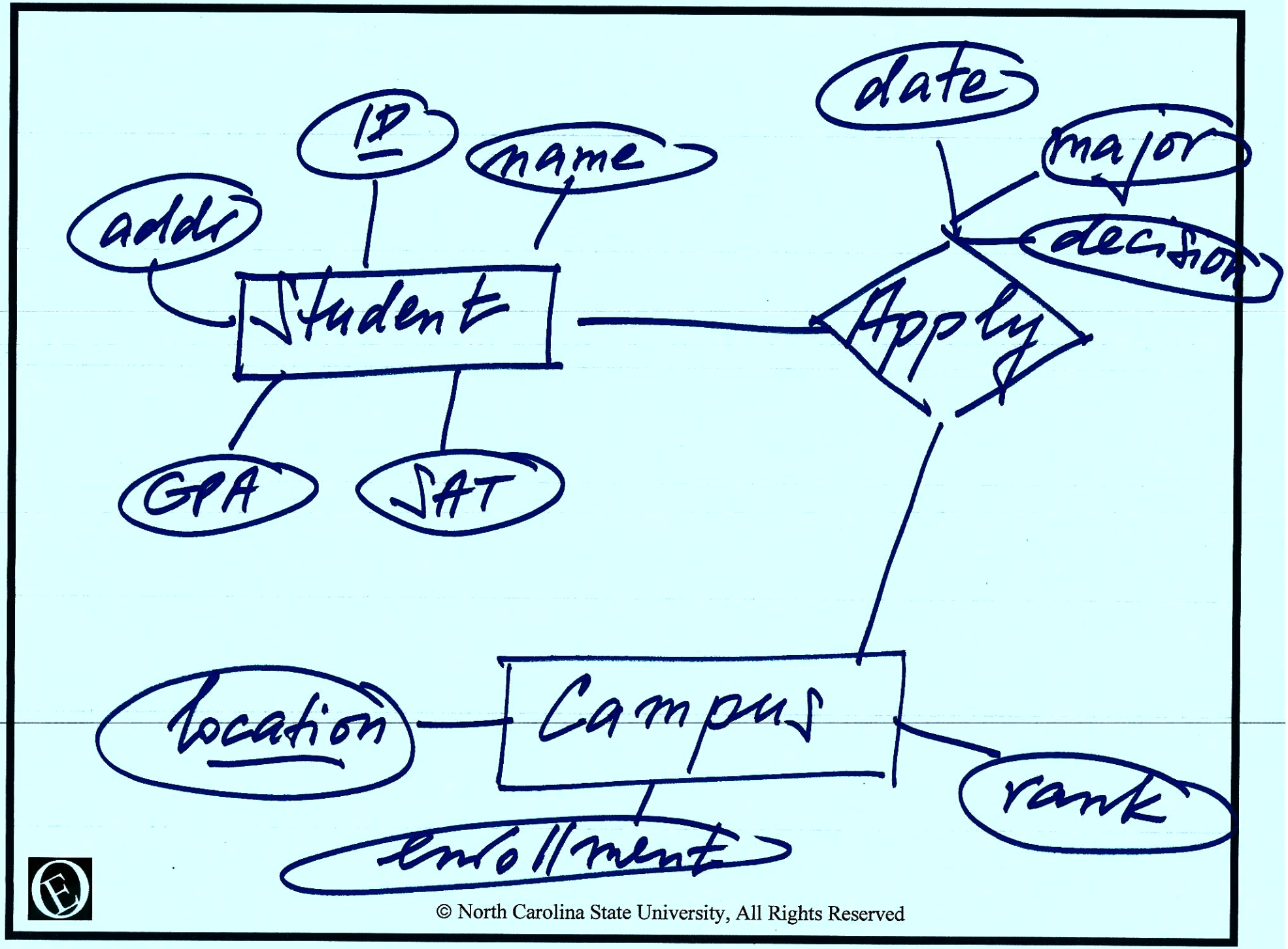
Triggers

* Integrity constraints: impose restrictions on the allowable data

* Have seen:
  - keys
  - functional dependencies
  - referential-integrity constraints (foreign-key constraints)

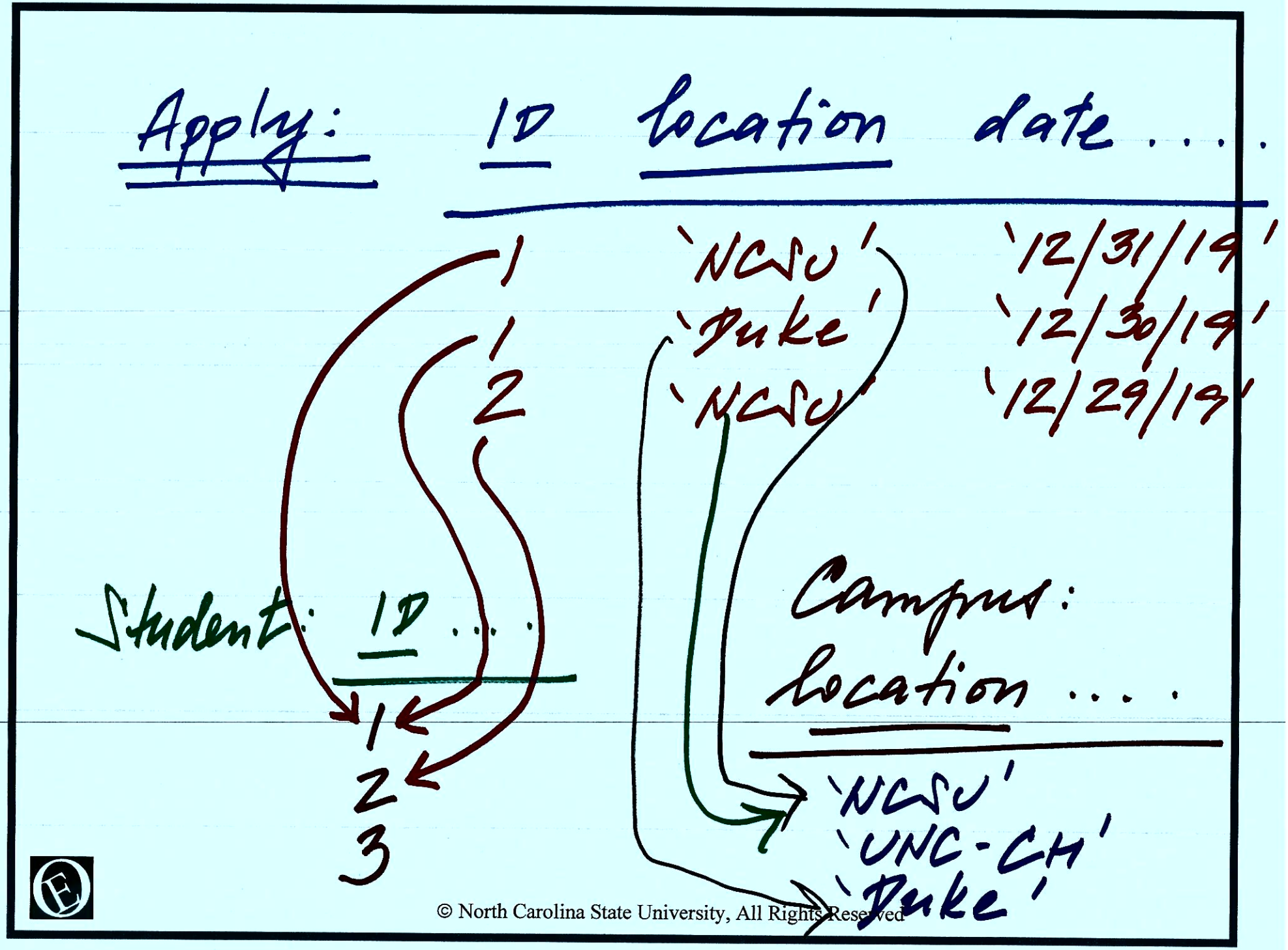
```
create table Student
    (ID integer primary key,
     name char(30) NOT NULL,
     address varchar(30),
     gpa float NOT NULL,
     sat integer,
     UNIQUE (name),
     UNIQUE (address));
```

```
create table campus
    (location char(25),
     enrollment integer,
     rank integer,
     primary key
         (location));
```



```
create table apply
    (ID integer references
                student(ID),
     location char(25),
     date char(10),
     major char(10),
     decision char,
     primary key (ID, location),
     foreign key (location)
                references Campus
                (location));
```

must be primary key or unique

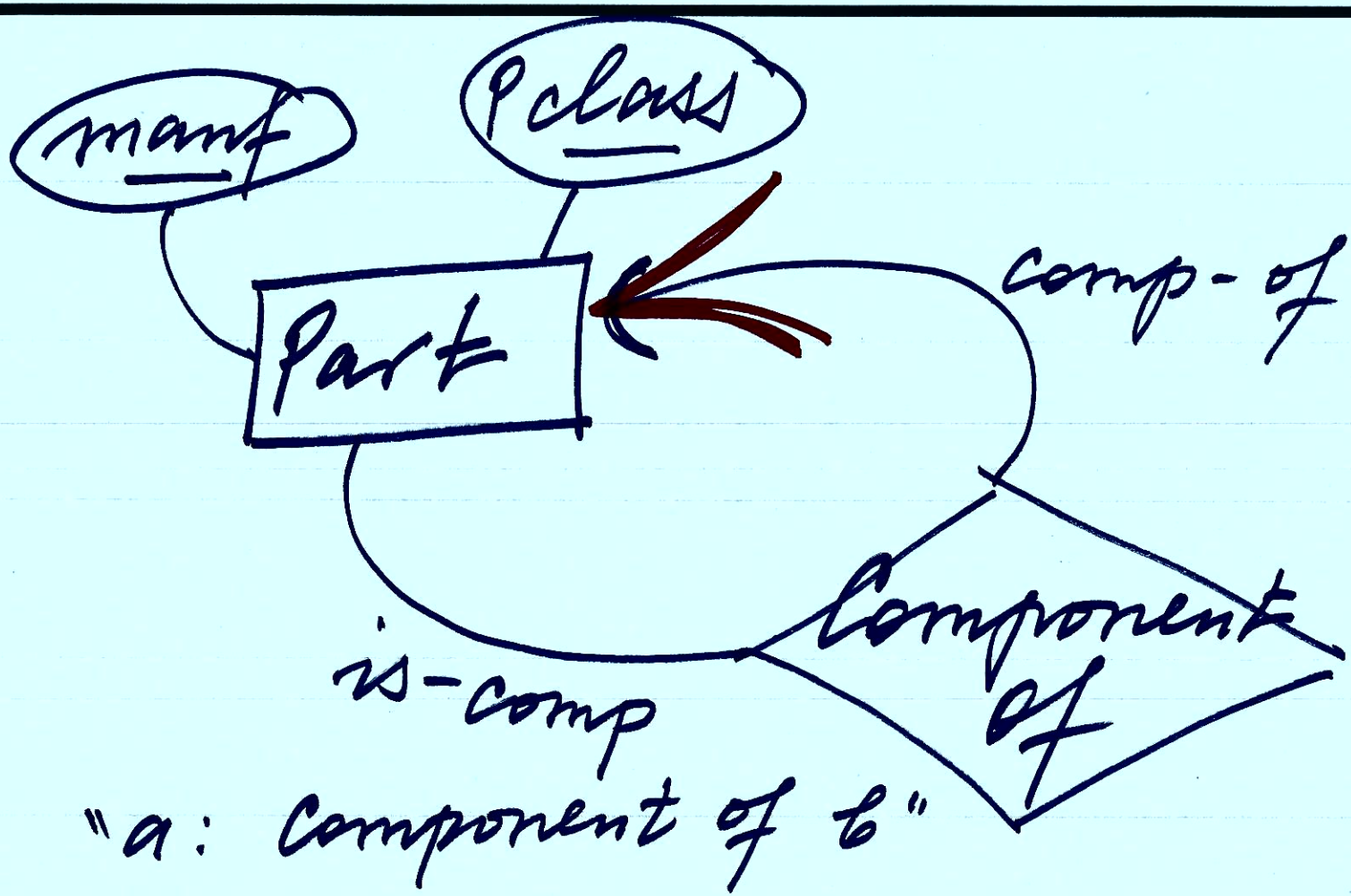


Referencing relation (e.g., Apply)

Referenced relation (e.g., Student)

1. Always disallow inserts/updates into "~~dangling~~ nonReferencable" values in Referencing relation

2. In the referenced relation: updates/deletions to "referenced" values can result in
- rejection (unless special declarations in CREATE TABLE) ~~by default~~
- [SET DEFAULT]
- SET NULL
- CASCADE



"a: component of b"

Part(manf, pclass, comanf, copclass)

```
create table Part
    (manf integer,
     pclass char,
     comanf integer,
     copclass char,
     primary key(manf,
                 pclass),
     foreign key(comanf,
                 copclass)
     references Part(manf,
                     pclass));
```

Part:

manf	pclass	comanf	copclass
1	'a'	NULL	NULL
2	'b'	1	'a'
3	'c'	2	'b'
1	'd'	3	'c'

1 a
↑
2 b
↑
3 c
↑
1 d

Attribute-based checks:

[Student]

GPA float
CHECK(GPA<=4.0
AND GPA>0),

[Apply]
decision char CHECK
(decision IN ('Y','N','U')),

```
[Apply_]
major char(10)
    CHECK (major NOT IN
        (SELECT
            deptName
        FROM Dept
        WHERE status = 'full'))
```