# DASC 5333 Database Systems for Data Science

## 1/30/2024 (self - annotation)

Welcome!

I am Bun Yue, the instructor.

Please take the time to check that the video and audio works for you in Zoom.

**5. A Simple Introduction to the Relational Model**

* Relational databases are the most popular databases: <https://db-engines.com/en/ranking>. It is based on the relational model.
* There are many other data models.
* In layman's term: A *table* (relation) is the basic unit of a relational database.
* A table is composed of many *rows*.
* Each row has many *column*values.
* A primary key is roughly a *minimal* set of columns in a table that*uniquely identify* a row.
* Two tables can be related to each other by *foreign keys*. A foreign key is roughly a column in a table in which its value must be equal to the referenced value of the primary key in another table (called the parent, referenced table).
* Microsoft's Access is based on the *relational model*. It may be considered as a desktop relational DBMS.
* Relational DBMS is the most popular DBMS. Examples:
  + DB-engine ranking: <https://db-engines.com/en/ranking>
  + Top 10 DBMS in Data Science: <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>
* SQL is the 'glue' in many DB systems.

**Introduction to  
Microsoft's Access**

by K. Yue

**1. Introduction to MS Access**

* MS Access is a relational database management system.
* MS Access is a very mature product and there are plenty of tutorials, examples, and resources.
* It is based on the *relational* model.
* To start, it is necessary to become familiar with the following basic concepts:
  1. table (or relation): contains rows of information
  2. row: contains information about a concept, event, object, entity, etc.
  3. column (or field, attribute) (or a row): stores the value of a property of a row.

A screenshot of a computer

Description automatically generated

* 1. column value
  2. data type: acceptable values of a column.
  3. data subtype

Learn the views of a graphical user interface (GUI).

A screenshot of a computer

Description automatically generated

* 1. primary key, PK (of a table): a set of columns that uniquely identify a row in a table.
     + No two rows should have the same value of PK.
     + Can a PK contains more than one columns? Yes

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

{stuId, classId} is the (composite) PK for the enroll table.

* 1. foreign key (of a table): a column that references a primary key of a A screenshot of a computer

     Description automatically generatedreferenced table.

A screenshot of a computer

Description automatically generated

* 1. relationship (between relation/table): usually refer to a foreign key referencing a primary key.
  2. relationship diagram (not to be confused with the entity relationship, or ER, diagram).
  3. query: an executable solution to a data problem.
  4. SQL: a standard query language for relational databases.
* As a GUI tool, there are various important *views*. MS Access includes the followings:
  1. Datasheet: allows insert/delete/update of individual rows.
  2. Query: design queries to answer questions using a Query by Example approach (Based on *Domain Relational Calculus*)
  3. Design: define tables, columns, keys, constraints, etc.
* There are also numerous database tools, e.g., relationship diagram.
* Our course requires a small percentage of the features provided by MS Access.
* As a starter, make sure you know how to:
  1. create a table
  2. populate a table
  3. create a simple query
* Many features are not covered in Access in this course, such as:
  1. create forms
  2. create reports
  3. Programming in MS Access
* In order to use MS Access, you may:
  1. Use a computer in the department labs or UHCL labs with MS Access installed.
  2. Gain MS Access license and install it in your computer. (However, Office 365 via UHCL does not include MS Access)
  3. Use UHCL virtual lab, Apporto: <https://www.uhcl.edu/computing/labs/virtual-lab>.

**2. Access Architecture**

The general database architecture uses a client-server architecture:

A diagram of a server and client architecture

Description automatically generated

1. A DB server, S, listens to a port.
2. A DB client, C, connects to the DB server and sets up a session.
3. C sends a SQL command to S.
4. S executes the SQL command and sends the result back to C.

MS Access contains both the DB client and DB server:

A diagram of a software server

Description automatically generated

1. MS Access Users use the Graphical User Interface (GUI) to develop queries.
2. SQL commands are generated from the GUI queries.
3. SQL commands are executed by the MS Access DB engine.
4. Results are displayed in Access GUI.

**3. Brief Introduction to the Relational Model**

The basic relational data model in layman terms:

1. A database is composed of a collection of*tables* (relations).
2. A table contains many *rows* (tuples, records) and *columns* (attributes, fields)
3. Each row contains many column values.
4. Every row of a table has the same columns.
5. Values of the same column have the same *data type*.
6. A table has a *primary key* to uniquely identify a row.
7. Data from two tables R1 and R2 can be linked by mean of a *foreign key* of R1 and a *primary key* of R2 (known as the *parent table*).

***Example:***

Consider [toyu.accdb](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/access/toyu.accdb):

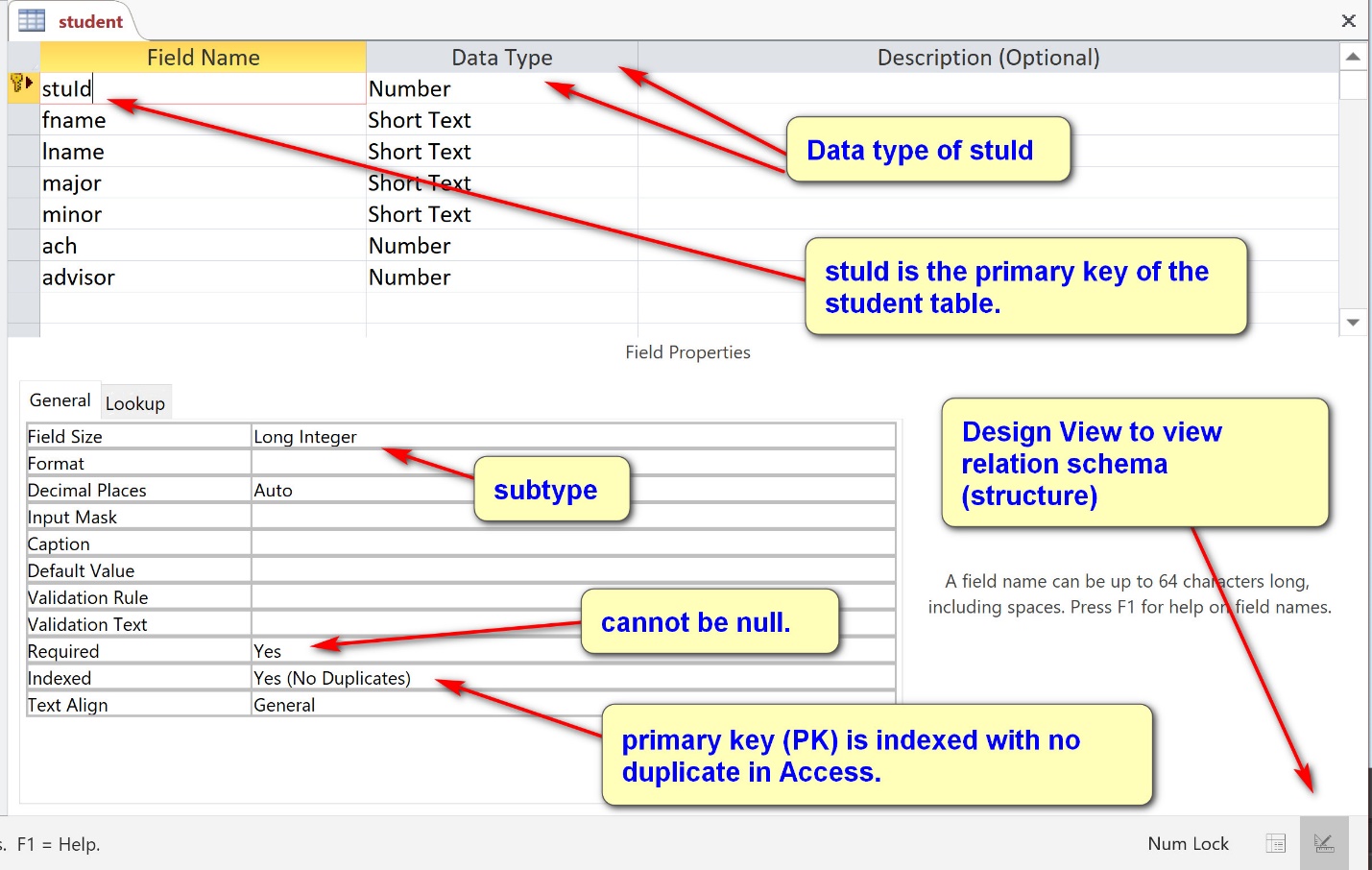
The database toyu has eight tables (primary keys are underscored):

1. grade(grade, gradePoint)
2. school(schoolCode, schoolName)
3. department(deptCode, deptName, schoolCode, numStaff)
4. faculty(facId, fname, lname, deptCode, rank)
5. course(courseId, rubric, number, name, credits)
6. class(classId, courseId, semester, year, facId, room)
7. student(stuId, fname, lname, major, minor, ach, advisor)
8. enroll(stuId, classId, grade, n\_alerts)

The student table contains 10 rows at the moment:



Student has seven columns:



Columns should be defined, such as in a *data dictionary*. For examples, for the table student:

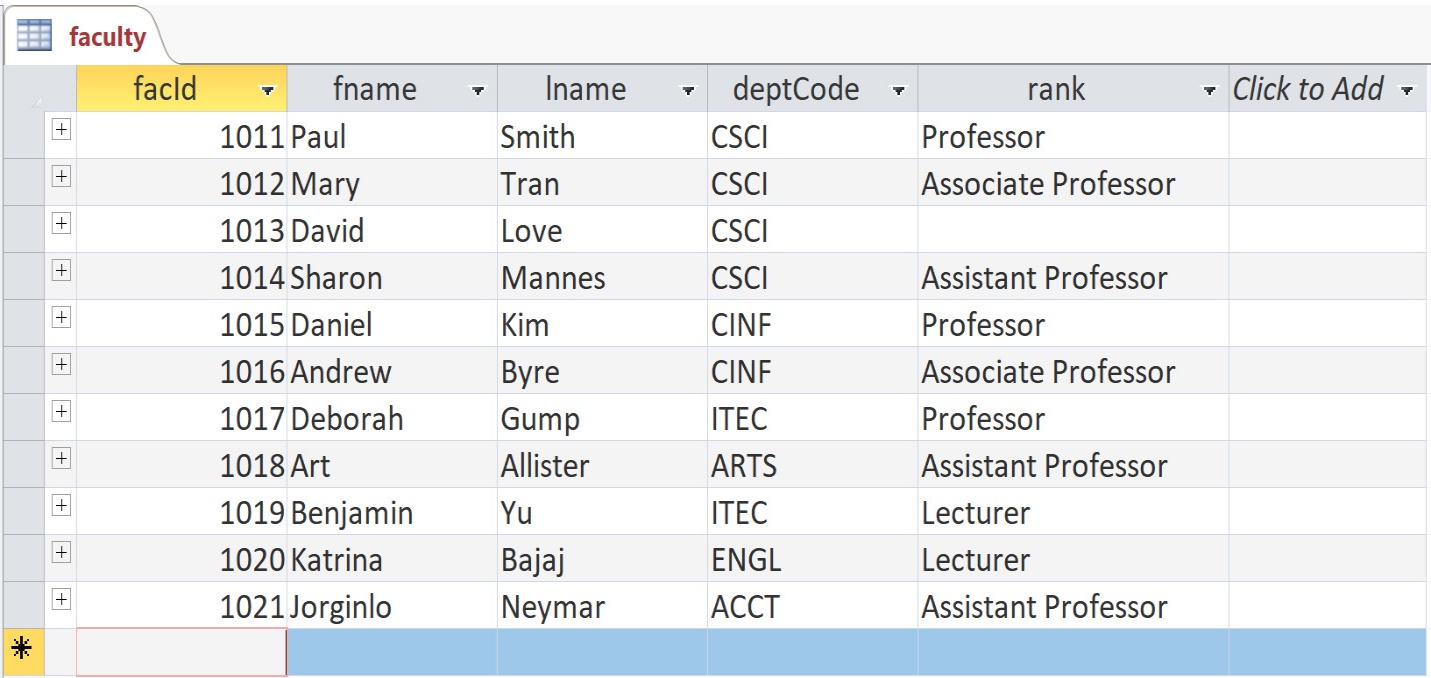
Student: a row in the student table stores the information of a student.

1. stuId: a unique id for the student (primary key).
2. fname: the first name of the student.
3. lname: the last name of the student.
4. major: the department code (deptCode) of the major of the student. Major may not be declared, and thus a null value is acceptable.
5. minor: the department code (deptCode) of the minor of the student. Minor may not be declared, and thus a null value is acceptable.
6. ach: the number of accumulated credit hours, including transferred credits.
7. advisor: the faculty id (facId) of the faculty who serves as the advisor of the student. A student may have no faculty advisor.

There are three foreign keys in the student table:

1. major references department(deptCode)
2. minor references department(deptCode)
3. advisor references faculty(facId)

The table faculty:



***Classroom exercise:***

Provide the *data dictionary* for the table faculty.

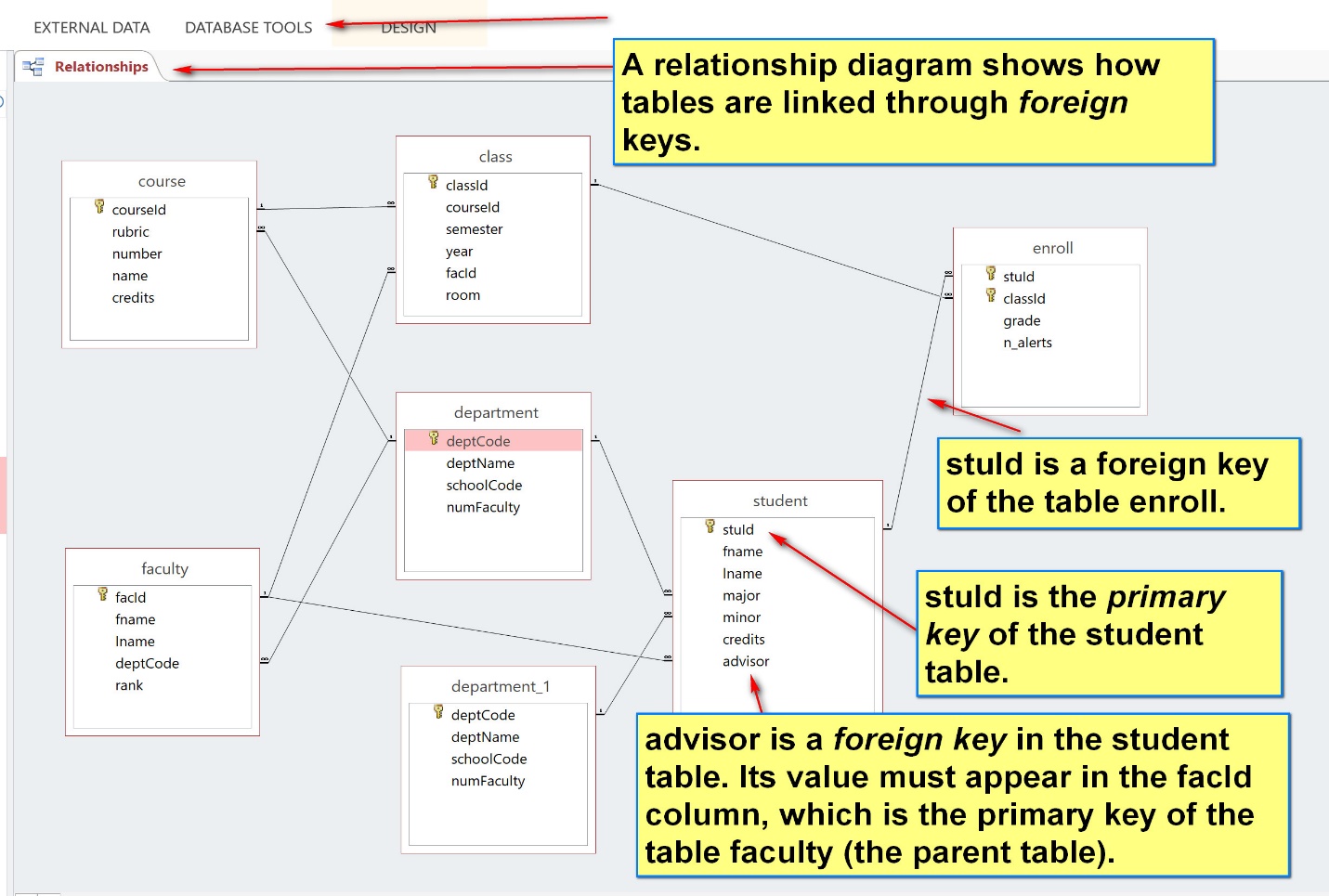
**4. Keys**

Consider the stuId 100000 in the student table:

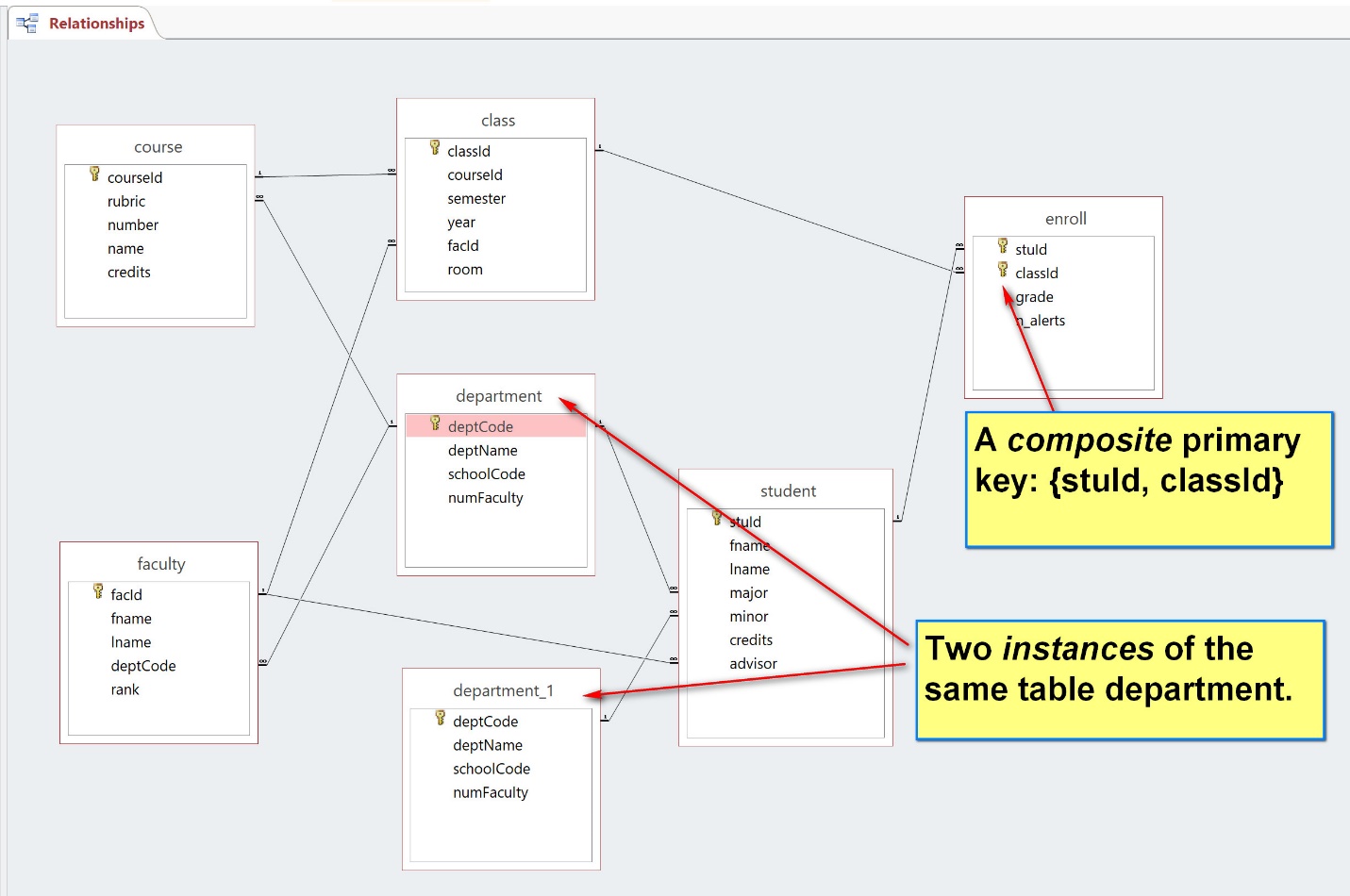
(stuId: 100000, fname: 'Tony', lname: 'Hawk', major:'CSCI', minor: 'CINF', credits:40, advisor:1011)

The column 'advisor' is a *foreign key*in the table student. Its value, 1011, must appear in the column 'facId', which is the *primary key* of the table faculty (the parent table).

A *relationship diagram* in Access shows foreign key relationships between *tables*.



A primary key may be a *composite key* (containing more than one columns) and a table can have multiple foreign keys.



**Classroom Demonstration or Practice Exercise**

(1) List the primary keys and foreign keys of all tables.

(2) Insert/Update/Delete

(a) Using toyu.accdb, add a new student:

(stuId: 100100, fname: 'Stephanie', lname: 'Smith', major:'MATH', minor: 'CINF', credits:33, advisor:1012)

(b) Lillian Johnson now has no minor.

(c) The class with id 11004 is canceled and its information should be deleted.

(3) New table: Create a new table Semester with the following columns:

1. SemesterId: auto number
2. Semester: 30 char string
3. Description: 255 char string

Insert the following four rows into the newly created table.

| **Semester** | | |
| --- | --- | --- |
| **SemesterId** | **Semester** | **Description** |
| 1 | Fall | Regular Fall semester |
| 2 | Spring | Regular Spring semester |
| 3 | Summer | Regular Summer semester |
| 4 | Fall First 8 weeks | Fall First 8 week semester |

(4) Foreign key:

In the class table, change the data type of "semester" to number. Rename the field to "SemesterId". Replace the field values of "Fall" and "Spring" to 1 and 2 respectively. Create a foreign key for the class table: SemesterId references Semester(SemesterId).

See [toyu\_withClasswork.accdb](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/access/toyu_withClasswork.accdb)

We will demonstrate how to construct queries to satisfy data problems in the class.

# Query

F2023

(a) Show the stuId, names and advisorId of all students in the following format. The expected result:

A screenshot of a computer

Description automatically generated with medium confidence

Declarative Analysis (in contrast algorithmic analysis):

[1] Output columns (label: value)

1. stuId: student.stuId
2. student: student.fname & ‘ ‘ & student.lname (expression: +: string concatenation.)
3. major: student.major
4. AdvisorId: student.advsior

[2] Source tables:

1. student

[3] Conditions: no condition

(c) Show the id and names of students who have received a grade of A or A- in a CSCI courses in the following format. The result:

A picture containing text, line, font, software

Description automatically generated

Note that Tony Hawk and Mary Hawk are the only students who have a grade of A or A- in some CSCI courses:

A screenshot of a computer

Description automatically generated with low confidence

Tips: you may need to change the property of the MS Access Query to show only distinct values.

Declarative Analysis (in contrast algorithmic analysis):

[1] Output columns (label: value)

1. stuId: student.stuId
2. student: [student].[fname] & ' ' & [student].[lname]

[2] Source tables:

1. student
2. enroll
3. class
4. course

[3] Conditions: who have received a grade of A or A- in a CSCI course

Problem conditions: who have received a grade of A or A- in a CSCI course: the same student contains a row in the enroll table with grade A or A-, with a class of CSCI course.

1. student.stuId a row in enroll table with grade A or A-
2. classId is a CSCI course:
3. course.rubric = ‘CSCI’

Join conditions (Foreign key = primary key of the parent table)

1. student.stuId = enroll.stuId
2. enroll.classId = class.classId
3. class.courseId = course.courseId