CS5095701 Advanced Database Systems
Homework 1

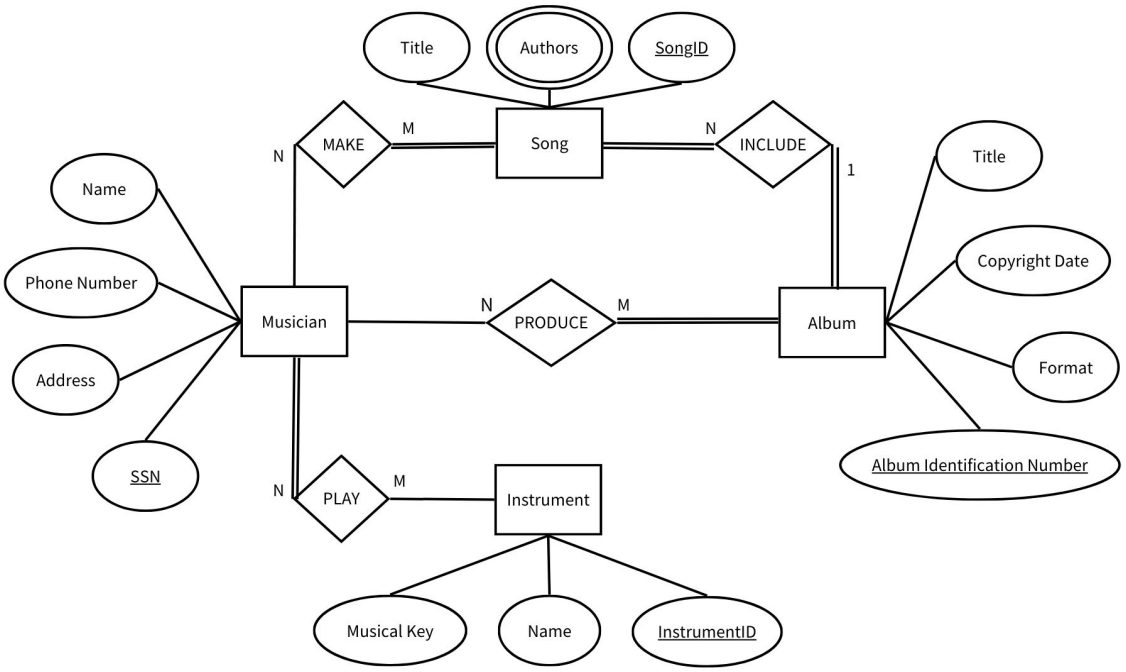
M11215032 葉品和

1. (70%) Following are the business descriptions written by domain experts who try to develop a database system for the Tower record company.

- Each song recorded at Tower has a title and one or more authors.
- Each album has one or more songs. But no song can appear on more than one album. Each musician that records at tower has an SSN, a name, an address, and a phone number. Musicians that are poorly paid may share the same address. No address has more than one phone.
- Each instrument used in song recorded at Tower has a name (e.g., violin, piano) and a musical key (e.g., B-flat, C-sharp).
- Each album has a title, a copyright date, a format (e.g., CD, vinyl), and an album identification number.
- Each musician may play several instruments, and a given instrument may be play by several musicians.
- Each song is performed by one or more musicians, and a musician may perform a number of songs.
- Each album can have one or more musicians who act as its producers. A musician may produce more than one album.

For the business description above, do the following.

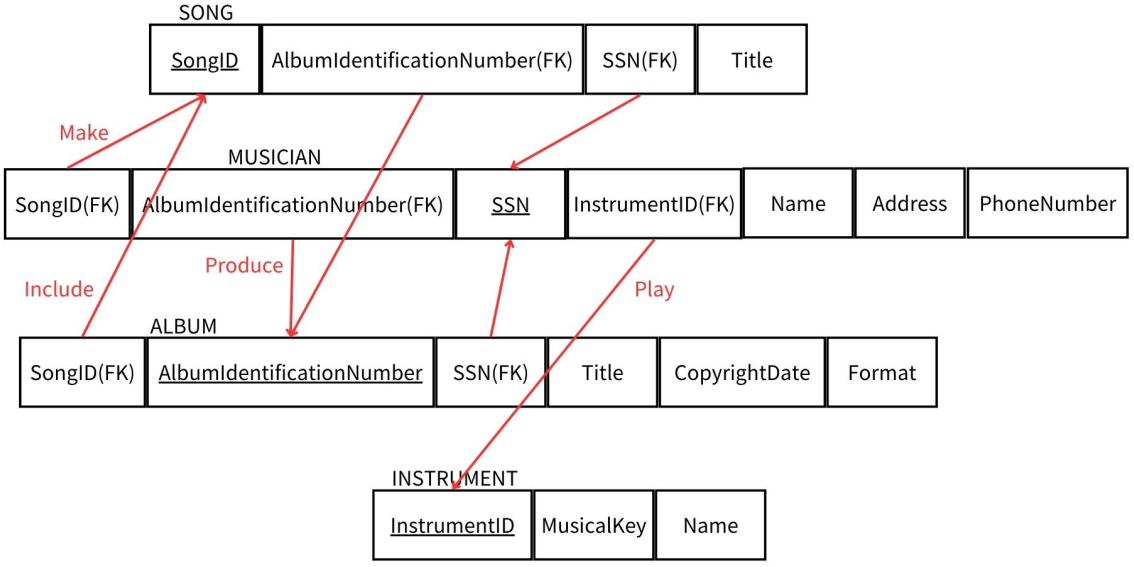
a. (15%) Draw the ER schema diagram.



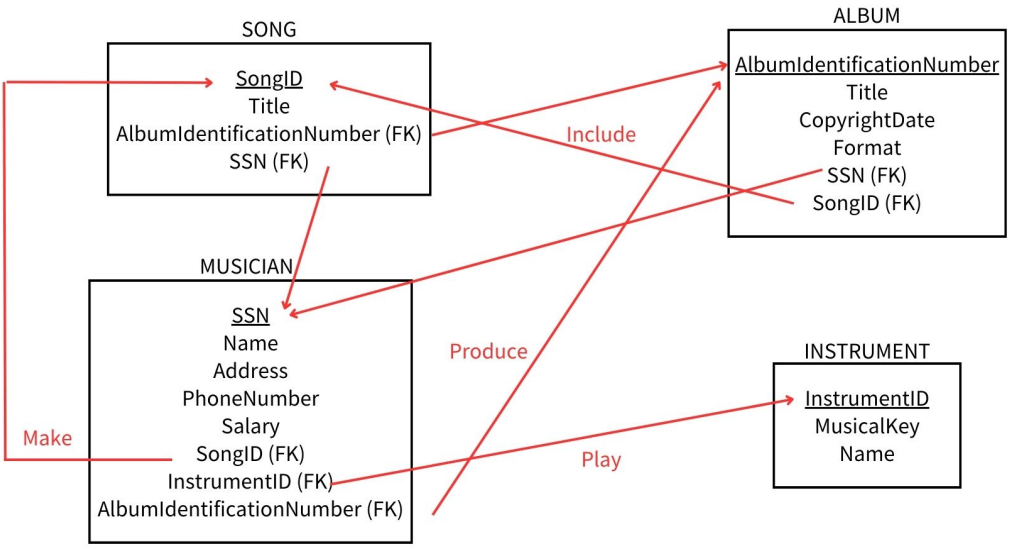
圖一、ER schema diagram

b. (15%) Convert your ER schema diagram from (a) into a relational schema.

我這邊畫了2種 relational schema，以橫式或直式的方法呈現。

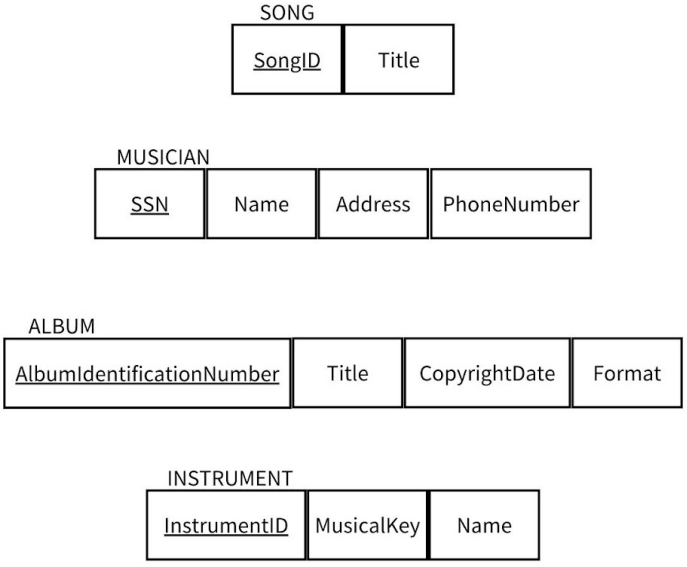


圖二、relational schema - 1



圖三、relational schema - 2

c. (15%) Normalize your relational schema from (b) to Boyce-Codd Normal Form. Justify your answers.



圖四、Boyce-Codd Normal Form

d. (25%) Write the query “Find the names of the albums with at least one song using the violin and in key C-sharp” in relational algebra, tuple relational calculus, domain relational calculus, and SQL.

- **relational algebra**

  $\{\pi_{\text{AlbumName}} (\sigma_{\text{Instrument = ‘Violin’ } \wedge \text{ Musical\_Key = ‘C\_Sharp’}} (\text{Album} \bowtie \text{Song}))\}$

- **tuple relational calculus**

  {a.AlbumName | Album(a) ∧ a.Song(Instrument = Violin ∧ Musical_Key = C_Sharp) >= 1}

- **domain relational calculus**

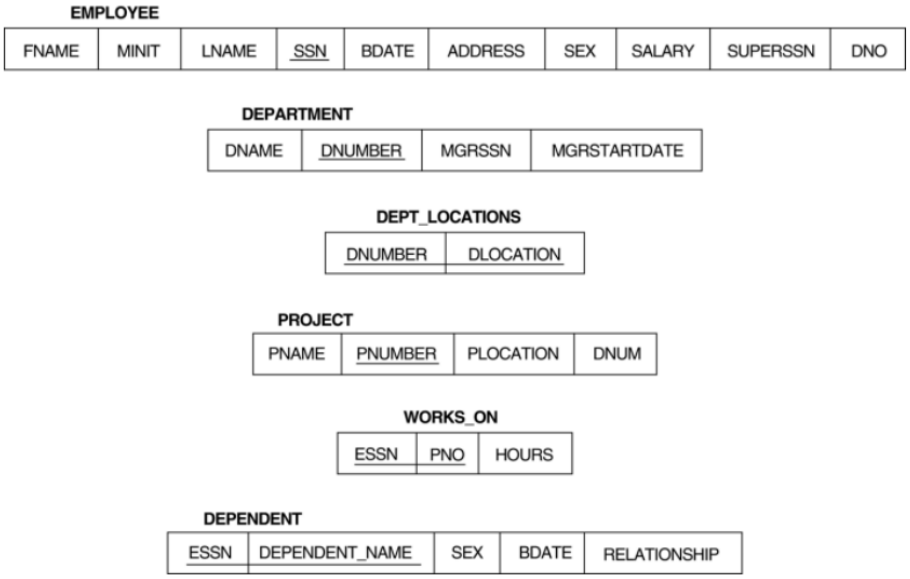
a: AlbumName	b: Song
c: Instrument	d: Musical_Key

  {a | ∃ (b) ∃ (c) ∃ (d) (Album(b, c, d) ∧ b(c = ‘Violin’ ∧ d = ‘C_Sharp’))}

- **SQL**

```
SELECT DISTINCT AlbumName
FROM Album
WHERE AlbumID IN (
        SELECT DISTINCT s.AlbumID
        FROM Song s
        WHERE s.Instrument = 'Violin' AND s.Musical_Key =
        'C_Sharp'
);
```

2. (30pt) Given the following relational schema:

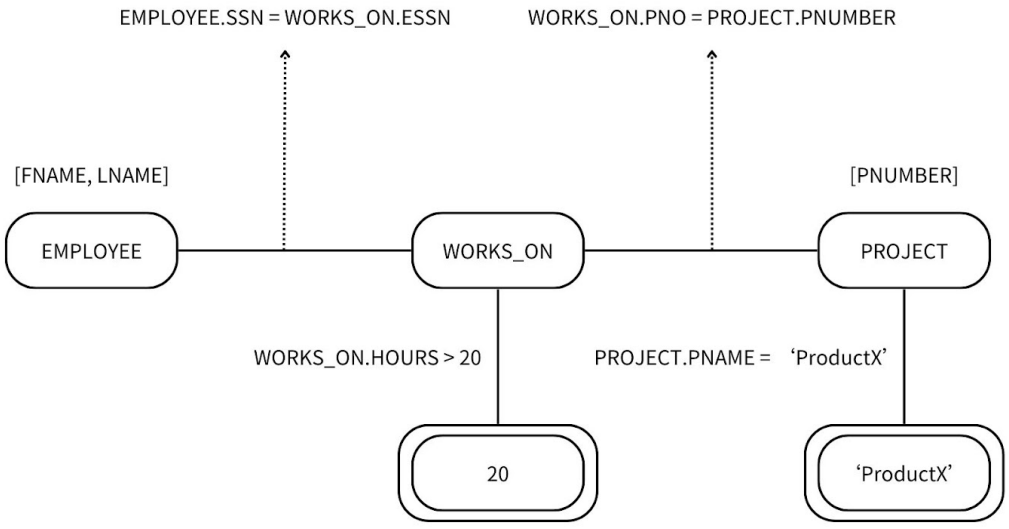


圖五、following relational schema

Consider the following SQL query

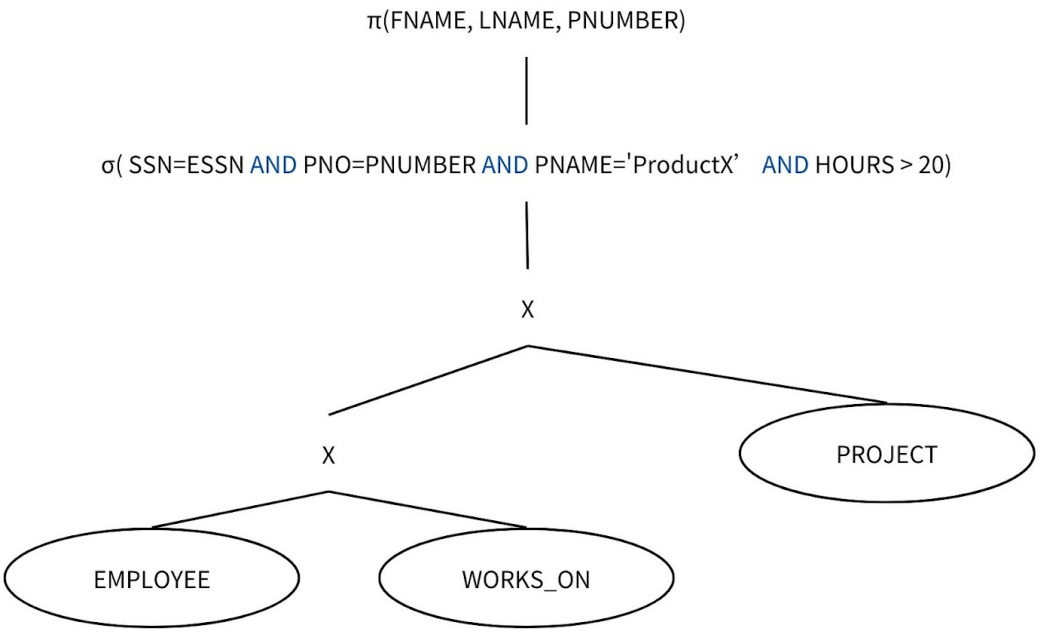
```
SELECT FNAME, LNAME, PNUMBER
FROM EMPLOYEE, WORKS_ON, PROJECT
WHERE SSN = ESSN AND PNO = PNUMBER AND PNAME = 'ProductX'
AND HOURS > 20
```

a. (15%) Draw the query graph.

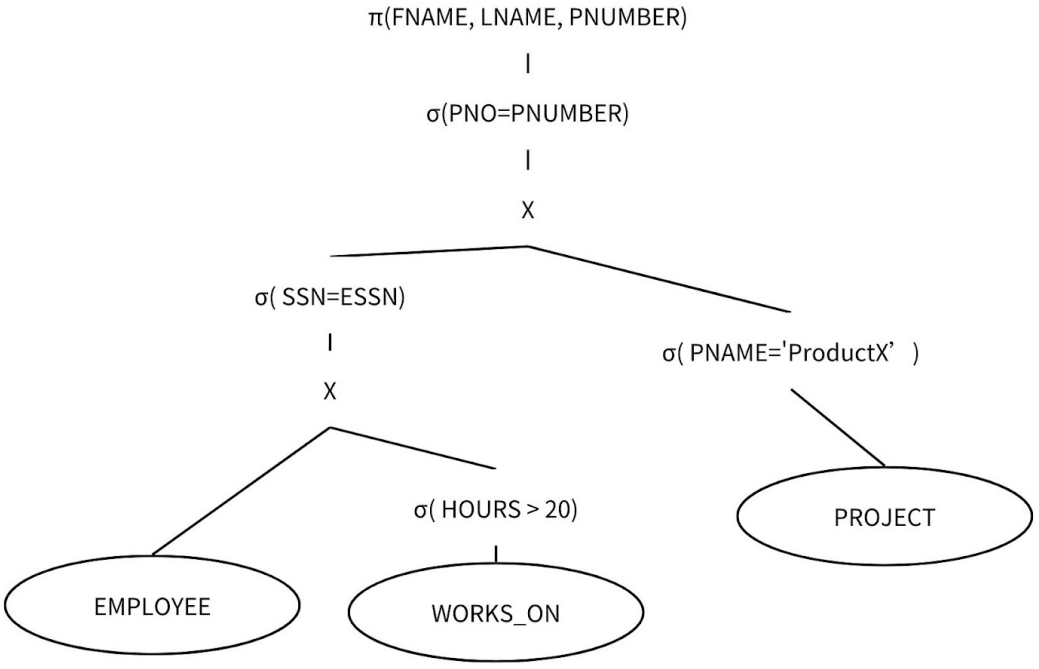


圖六、query graph

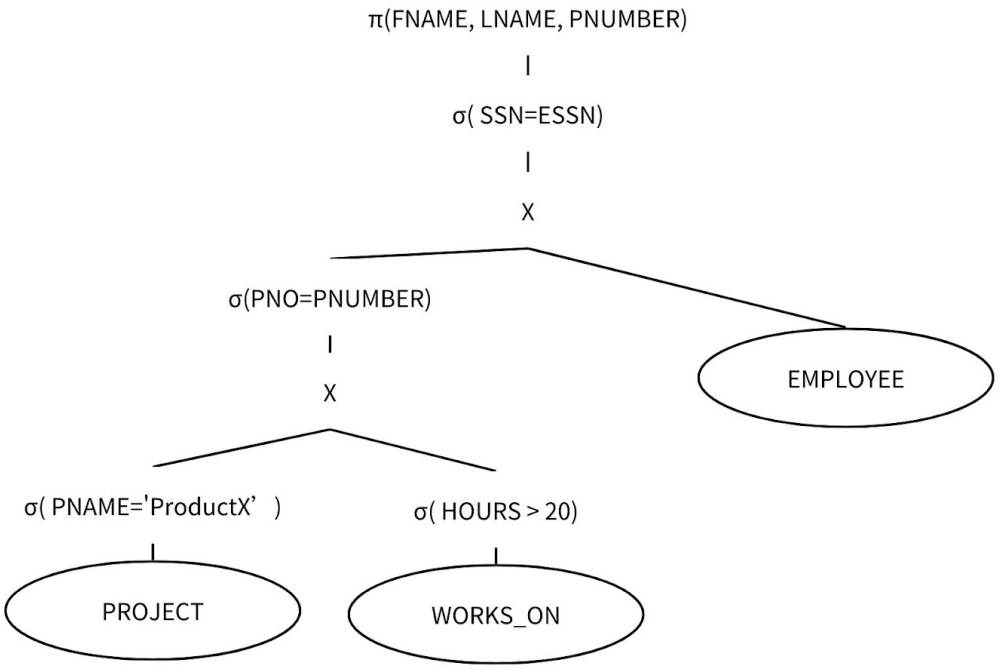
b. (15%) Draw the query tree that is optimized by the heuristic optimization outlined in class.



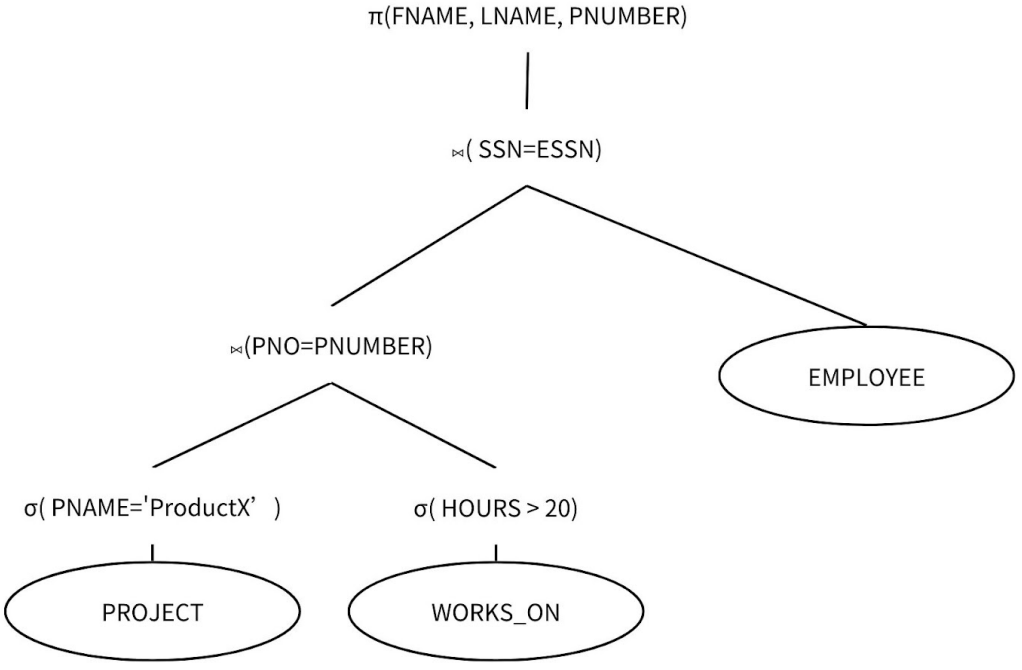
圖七、Initial (canonical) query tree for SQL query Q.



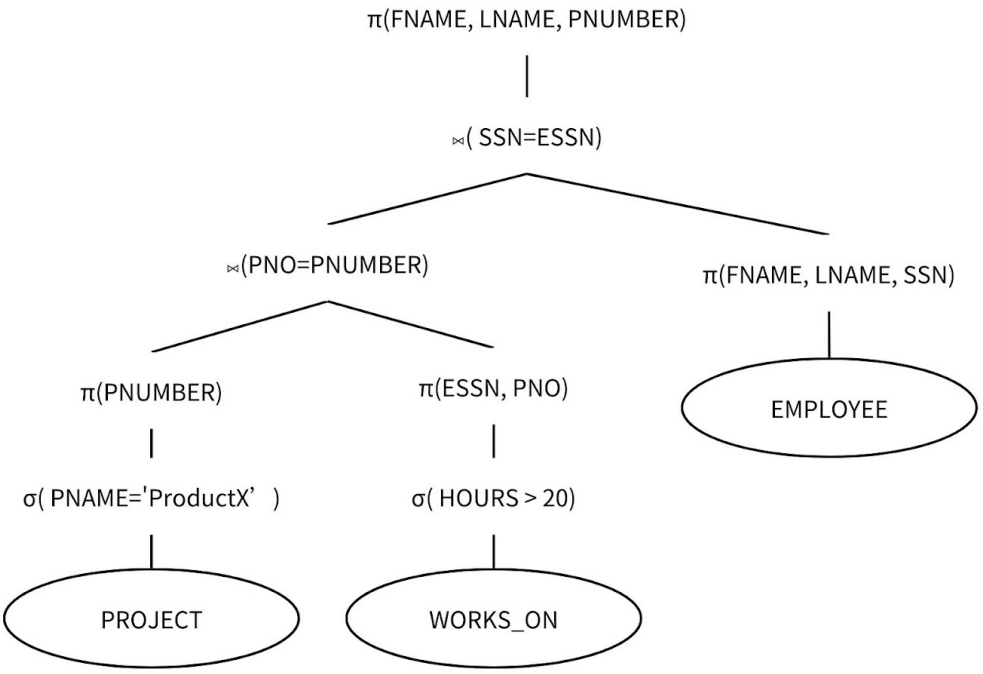
圖八、Moving SELECT operations down the query tree.



圖九、Applying the more restrictive SELECT operation first.



圖十、Replacing CARTESIAN PRODUCT and SELECT with JOIN operations.



圖十一、Moving PROJECT operations down the query tree.