Trivial:右邊之相依因素，為左邊決定因素的部份集合
Non-Trivial:右邊之相依因素，不是左邊決定因素的部份集合

Full Functional Dependency:
1.主鍵是多個屬性組成，某非鍵值屬性依賴主鍵之全
部，則稱該非鍵值屬性“完全相依”於主鍵。
2.若主鍵僅由一個屬性所組成，則任一非鍵值屬性必“完全相
依”於主鍵

。
Partial Functional Dependency:
若主鍵是多個屬性組成，而某非鍵值屬性是依賴主鍵之
部分屬性時，則稱該非鍵值屬性“部份相依”於主鍵。
Transitive Dependency:
若存在一個非鍵值屬性 T，使得 I→T 且 T→J 的功能相依
性均成立，則稱之 J 遞移相依於 I。

1NF: 除去非基元性(Non-atomic) atomic)資料項目
2NF: 除去部份功能相依性，滿足 1NF 且每一個非鍵屬性完全相依於主鍵
3NF: 除去遞移相依性，滿足 2NF 且每一個非鍵屬性非遞移相依於主鍵
BCNF: 除去其它功能相依性所引起的異常，每一個決定因素(Determinant)皆是 (Determinant) 候選鍵

Prime attribute: An attribute that is member of the primary key K

Full functional dependency: a FD Y -> Z where removal of any attribute from Y means the FD does not hold any more

Functional dependencies (FDs)
Are used to specify formal measures of the "goodness" of relational designs
And keys are used to define normal forms for relations
Are constraints that are derived from the meaning and interrelationships of the data attributes

Normalization:
The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

Normal form:
Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

Query optimization:
The process of choosing a suitable execution strategy for processing a query
Acid

- Atomicity（原子性）：一個事務（transaction）中的所有操作，或者全部完成，或者全部不完成，不會結束在中間某個環節。事務在執行過程中發生錯誤，會被回滾（Rollback）到事務開始前的狀態，就像這個事務從來沒有執行過一樣。即，事務不可分割、不可約簡。
- Consistency（一致性）：在事務開始之前和事務結束以後，資料庫的完整性沒有被破壞。這表示寫入的資料必須完全符合所有的預設約束、觸發器、級聯回滾等。
- Isolation（隔離性）：資料庫允許多個並發事務同時對其數據進行讀寫和修改的能力，隔離性可以防止多個事務並發執行時由於交叉執行而導致數據的不一致。事務隔離分為不同級別，包括未提交讀（Read uncommitted）、提交讀（read committed）、可重複讀（repeatable read）和串行化（Serializable）。
- Durability（持久性）：事務處理結束後，對數據的修改就是永久的，即便系統故障也不會丟失。

Serial schedule over a set S of transaction
Transaction之間的存取順序必須按照當初transaction開始的順序來操作data item（需要補全）
 2. (4%) Durability ACID properties
isolation ACID properties：

獨立性，各transcation之間獨立，不互相干擾

3. (4%) B+ Tree

https://zh.wikipedia.org/zh-cn/B%2B%E6%A0%91

B+ 树是一种树数据结构，通常用于数据库和操作系统的文件系统中。B+ 树的特点是能够保持数据稳定有序，其插入与修改拥有较稳定的对数时间复杂度。B+ 树元素自底向上插入，这与二叉树恰好相反

4. (4%) Two-Phase Locking Techniques

In databases and transaction processing,**two-phase locking**(2PL) is a concurrency control method that guarantees serializability.

5. (4%) Correlated Nested Queries (Hint: SQL)

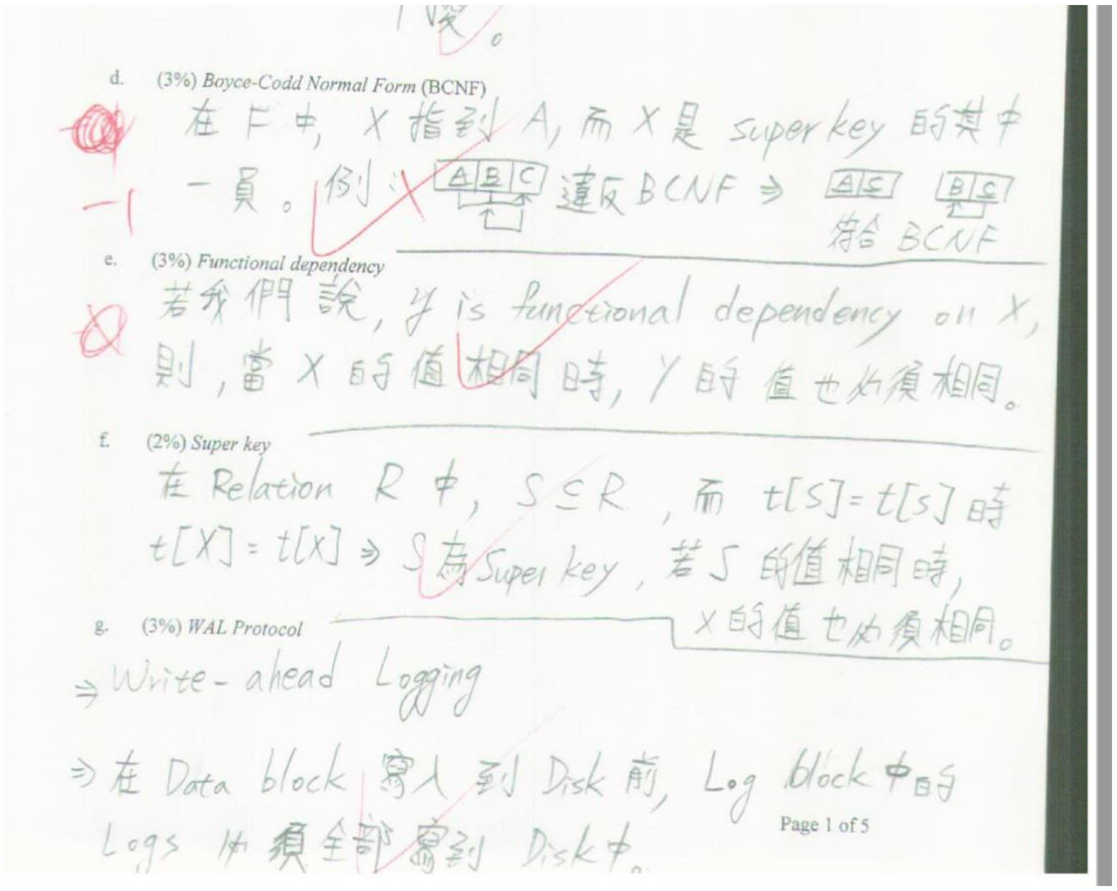
在 SQL数据库查询中，**相关子查询**（也称为**同步子**查询）是使用外部查询中的值的子查询（嵌套在另一个查询中的查询）。因为可以针对外部查询处理的每一行评估子查询一次，所以它可能是低效的。

以下是典型相关子查询的示例。在此示例中，目标是查找薪水高于其部门平均水平的所有员工。

https://en.wikipedia.org/wiki/Correlated_subquery

Strict 2pl

目前有拿到 write lock 的 transaction 不把它 release，直到此 transcation commit or abort



[illegible]。

d. (3%) *Boyce-Codd Normal Form* (BCNF)

-1

在 F 中，X 指到 A，而 X 是 super key 的其中一員。例：A B C 違反 BCNF ⇒ A C B C 符合 BCNF

e. (3%) *Functional dependency*

若我們說，Y is functional dependency on X，則，當 X 的值相同時，Y 的值也必須相同。

f. (2%) *Super key*

在 Relation R 中，S ⊆ R，而 t[S] = t[s] 時 t[X] = t[x] ⇒ S 為 Super key，若 S 的值相同時，X 的值也必須相同。

g. (3%) *WAL Protocol*

⇒ Write-ahead Logging

⇒ 在 Data block 寫入到 Disk 前，Log block 中的 Logs 必須全部寫到 Disk 中。

Page 1 of 5

1. 正規化

What teacher said:

normalize:
- a, b, c, d func dependcy
func key

ident normal form
decompose

- Similar problem:
  refer to hw4 no.1

2Query Processing

- What teacher said:
  given query, initial query tree, optimize, (give reason)
- ppt:
  refer to the ppt 18

3Recovery

given transaction log
given time point: system crash
- give what time need what log, how to recover

you may refer to ppt chapter 20