

Normalisation

To remove undesired redundancy from databases (Break one large table into several smaller tables)

Definition: A relation is normalized if all determinants are candidate keys

Problem: Anomalies in Denormalized Data

Student-ID	Course-ID	Fee
130	C200	75
200	C300	100
250	C200	75
425	C400	150
500	C300	100
575	C500	50
...

Insertion Anomaly: A new course cannot be added until at least one student has enrolled

Deletion Anomaly: If student 425 withdraws, we lose all record of course C400 and its fee

Update Anomaly: If the fee for course C200 changes, we have to change it in multiple records (rows), else the data will be inconsistent.

Functional Dependency

Definitions

$$A(\underline{X}, \underline{Y}, Z, D)$$

Determinants: The left hand side of the arrow, e.g. X, Y in $(X, Y \rightarrow Z)$

Key and Non-Key attributes: whether the attribute is part of the primary key or not

Partial functional dependency: a functional dependency of one (or more) non-key attributes upon part but not all of the primary key, e.g. $(Y \rightarrow Z)$

Transitive dependency: a functional dependency between 2 (or more) non-key attributes, e.g. $(Z \rightarrow D)$

Armstrong's Axioms

Functional dependencies can be identified using Armstrong's Axioms

$$A = (X_1, X_2, \dots, X_n) \text{ and } B = (Y_1, Y_2, \dots, Y_n)$$

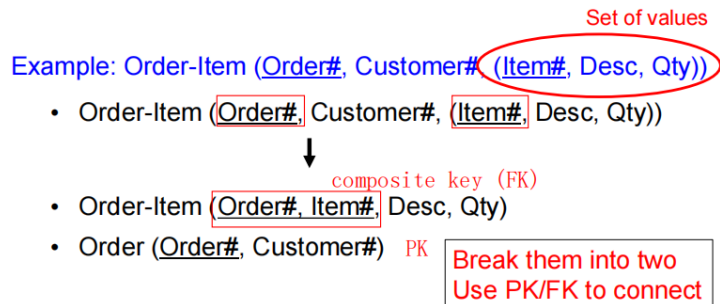
1. **Reflexivity:** $B \subseteq A \Rightarrow A \rightarrow B$ (自包含)
e.g. Student_ID, name \rightarrow name
2. **Augmentation:** $A \rightarrow B \Rightarrow AC \rightarrow BC$
e.g. Student_ID \rightarrow name \Rightarrow Student_ID, surname \rightarrow name, surname
3. **Transitivity:** $A \rightarrow B \text{ and } B \rightarrow C \Rightarrow A \rightarrow C$
e.g. ID \rightarrow birthdate and birthdate \rightarrow age \Rightarrow ID \rightarrow age

Normalisation

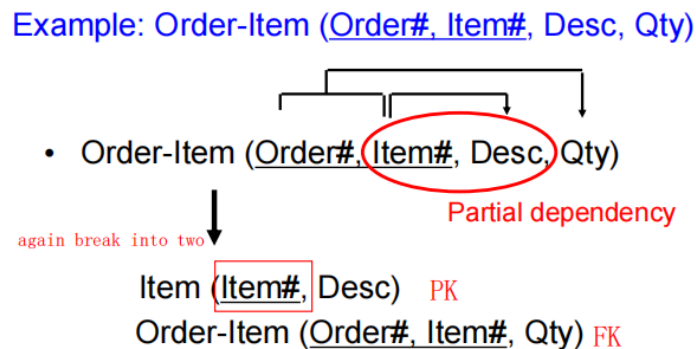
Steps in Normalisation

1. **First Normal Form**-Keep atomic data/Remove repeating groups

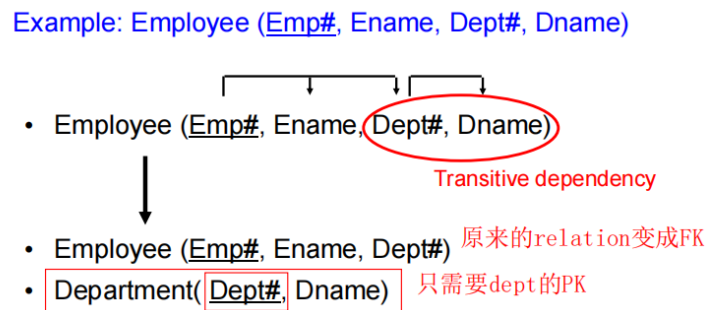
原PK r epeating PK (e.g. 含有多个value的列) -> CPK (composite FK) + New separate table
=> update, delete and insert anomalies still exists

2. **Second Normal Form**-Remove partial dependencies

breakdown + FK => update, delete and insert anomalies still exists

3. **Third Normal Form**-Remove transitive dependencies

breakdown + FK => no anomalies!



Normalisation vs Denormalization

Normalisation:

1. Contains a minimum amount of redundancy
2. Allow users to insert, modify, and delete rows in tables without errors or inconsistencies (anomalies)

Denormalization:

1. Specific Query Speed is super fast
2. Extra work on updates to keep redundant data consistent
3. Denormalization may be used to improve performance of time-critical operations