

Normalisation

Student Name: Aishwarya Manumari

- a. Relations that contain redundant information may potentially suffer from update anomalies. The three types of update anomalies include insertion, deletion, update.

1. Insertion Anomalie

- To insert the details of new member of Staff into the StaffHotel relation, the details of the Hotel at which the staff are located must be included.

For example, if new staff member John is located at hotel H25, East Kilbride, the correct details of contractNo must be entered so that the Hotel details are consistent with values for C1024 in other tuples of the StaffHotel relation. However if the new staff member John located at hotel H24, London is inserted into the StaffHotel relation will cause potential inconsistency, because that contractNo doesn't exist yet thus no hotel with that contractNo is located, therefore entering only the appropriate contractNo for each staff member in the Staff relation prevents inconsistency.

For example: Insert a new Staff member

NIN	contractNo	hours	eName	hNo	hLoc
1128	C1024	12	John S	H25	East Kilbride

- To insert details of a new hotel that currently has no staff members into the StaffHotel relation, it is necessary to enter null values into the attributes for Staff, ie. NIN, eName. Since NIN is the primary key it cannot be null as it violates entity integrity rules. Hencefore having Staff and Hotel relation's separate is appropriate.

2. Deletion Anomalie

Deleting a tuple from the StaffHotel relation ex: last member of staff with C1023 located at a hotel, the details about that contract with hotel information are also lost from the database. For ex: if we delete tuple for staff NIN 1130 with contract C1023 located at hotel H24, London is removed from the database. This becomes an issue as contractNo of that employee is deleted with hotel details where hNo, hLoc are dependent on contractNo in the StaffHotel relation. This problem can be avoided when hotel tuples are stored separately from staff tuples. This ensures if staff NIN 1130 is deleted from the Staff relation, the details on contractNo, hNo, hLoc remain unaffected in the Hotel relation.

3. Modification/ Update Anomalie

An Update anomaly ensure all details are changed for each tuple in a relation. For ex: if the hotel location for hotel number H25 was to be changed to London, all tuples must be updated of all staff located at that hotel number. This update must happen in all appropriate tuples of the StaffHotel relation to avoid inconsistency. To avoid these anomalies and resolve dependency, decomposing the original relation into the Staff and Hotel relations is appropriate.

b. 1NF

StaffHotel relation has four functional dependencies. The determinants in StaffHotel relation are:

$NIN \rightarrow eName$ (fd1)

$ContractNo \rightarrow hNo, hLoc$ (fd2)

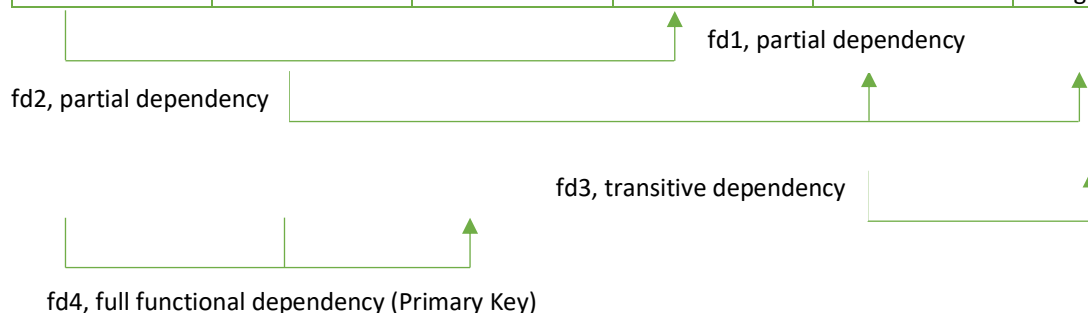
$hNo \rightarrow hLoc$ (fd3)

$NIN, ContractNo \rightarrow hours$ (fd4)

The only determinant that functionally determines all the other attributes of the relation is (NIN, ContractNo) (NIN, ContractNo) - is identified as the primary key for StaffHotel relation.

StaffHotel Relation

NIN	contractNo	hours	eName	hNo	hLoc
1135	C1024	16	Smith J	H25	East Kilbride
1057	C1024	24	Hocine D	H25	East Kilbride
1068	C1025	28	White T	H4	Glasgow
1135	C1025	15	Smith J	H4	Glasgow



Normalisation

Student Name: Aishwarya Manumari

- c. The **StaffHotel** relation is currently in 1NF, in which the intersection of each row and column has only single atomic value (per cell). Repeated data along with a copy of the original key attributes is inserted as a separate relation per tuple.

1NF to 2NF

StaffHotel relation is in 1NF where NIN, contractNo, is a composite primary key where NIN and contractNo are the primary keys. NIN determinant of hours is partially dependent as subset of the primary key also functionally defines an attribute in the relation, contractNo \rightarrow hours. Therefore a partial dependency exists on the primary key. Partial dependency on the primary key (NIN, contractNo) is removed by placing them in a new relation along with a copy of their determinant.

2NF

Staff

fd1 NIN \rightarrow eName (Primary Key)

Hotel

fd2 contractNo \rightarrow hNo, hLoc (Primary Key)

StaffHotel

fd4 NIN, ContractNo \rightarrow hours (Primary Key)

2NF to 3NF

Transitive dependency in **StaffHotel** relation exists where the primary key is contractNo. It explains the condition where A, B and C are attributes of a relation such that A is determinant of B and B is determinant of C, where A is the primary key. contractNo and hotelNo are attributes of a relation such that contractNo is functionally determinant of hotelNo and hotelNo is determinate of hotelLoc. hotelLoc is transitively dependent on contractNo through hotelNo. Thus, transitive dependency on the primary key contractNo is removed by placing them in a new relation along with a copy of their determinant.

3NF

Staff

fd1 NIN \rightarrow eName (Primary key)

StaffHotel

fd4 NIN, ContractNo \rightarrow hours (Primary Key)

Hotel

fd2 contractNo \rightarrow hNo (Primary Key)

fd2 hNo \rightarrow hLoc (Transitive dependency)

hNo is transitively dependent on contractNo.

Staff (NIN, eName)

Primary Key (NIN)

Hotel (contractNo, hNo, hLoc)

Primary Key (contractNo)

StaffHotel (NIN, contractNo, hours, eName, hNo, hLoc)

Primary Key (NIN, ContractNo)

Foreign Key (NIN) **references** **Staff** (NIN)

Foreign Key (ContractNo) **references** **Hotel** (ContractNo)