

Database



**ASSIGN
BUSTER**

Database Name: Course: Date: Database #16. 20 The third normal form (3NF) of database relations specifies that references to columns, which are not dependent on the primary key, be eliminated. 3NF therefore requires the use of foreign keys to reference other separate tables.

No columns in the referenced table should originate from the parent table. The relation schemas produced by Algorithm 16. 6 can be in 3NF if they satisfy the following conditions. The relational table (R) must be in 2NF or 1NF. All columns should be fully dependent on the primary key. To determine whether the relation schemas are actually in 3NF we will have to prove that they satisfy the above conditions. The algorithm in 16.

6 comprises of Universal Relation R and a set of FDs (Functional Dependencies) F that are attributes of the universal relation. These FDs are in the form of sets R1, R2 and R3 and are given as follows: R1 (Empssn, Esal, Ephone, Dno) R2 (Pno, Pname, Plocation) R3 (Empssn, Pno) R1 is a functional dependency of the relation schema that contains a set of attributes, which include; the Employee social security number (Empssn), employee salary (Esal), employee phone number (Ephone) and the (Dno). Therefore R1 or FD1: Empssn > Esal, Ephone, Dno. The same is applied to the other functional determinants as follows: R2 or FD2: Pno > Pname, Plocation R3 or FD3: Empssn > Pno For explanation purposes, the above attributes will be abbreviated with the first one or two letters in the case where attributes share the first letter.

For example, Empssn will be Em, so the functional dependencies can be merged to form the universal set R as represented below; F: {Em > EsEpD,

$Pno > PnPI, Em > Pno$ } F therefore represents the complete relation schema with all the attributes incorporated from the FDs. To prove that this relation schema is in the third normal form we first identify whether it is in second normal form. F is in Third normal form because there is a trivial dependency $Em > Pno$. This satisfies the requirement of 2NF. The second requirement is to identify if the non-prime attribute is dependent on the super keys of R. The Employee social security number (Empno) is the super key in this universal relation.

The Pno also acts as a key in one of the functional determinants (i. e. R2).

The Pno is the non-prime attribute. Pno exhibits functional dependency in which the Empno indirectly determines the Pno. # 16. 21 Template dependencies are methods used to represent constraints among relation databases that are difficult to describe. This is done by describing the constraints using a template. In our case, the template will be made of hypothesis tuples and conclusion.

The relation schema is made up of functional dependencies that can be represented in a dependency template as shown below. R: {Em, Pno, Es, Ep, D, Pn, PI} Hypothesis: $Em_1 Pno_1 Es_1 Ep_1 D_1 Pn_1 PI_1$ $Em_2 Pno_2 Es_2 Ep_2 D_2 Pn_2 PI_2$ X = {Em, Pno} Y= {Es, Ep, D, Pn, PI} The above template is made up of several tuples. This template shows that if the tuples $Es_1 \dots PI_1$ (the antecedents) appear in the database, then the corresponding tuples $Es_2 \dots PI_2$ must also appear in the same database.

Conclusion: $Es_1 = Es_2, Ep_1 = Ep_2, D_1 = D_2, Pn_1 = Pn_2, PI_1 = PI_2$ The above templates show the template dependencies among various tuples in the

relation schema. The employee number is the super key and all subsequent tuples are dependant on the super key. The relation between these attributes is illustrated using the template above. #16. 22 Relation schema 3. 5 has several inclusion dependencies. This relation schema (R) has the following sets of attributes; Employee, department, dept_locations, project, works_on and dependent which will be given as R1, R2, R3, R4, R5 and R6 respectively. Therefore; R1 :{ Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno} R2: { Dname, Dnumber, Mgr_ssn, Mgr_start_date } R3: {Dnumber, Dlocation } R4: { Pname, Pnumber, plocation, Dnum } R5: { Essn, Pno, Hours } R6: { Essn, Dependent_name, Sex, Bdate, Relationship } To find out the cases of inclusion dependency we must follow the following rules; Reflexivity: $X < X$, in this case, there must be a functional dependency between R.

X and its subsets. In this case, R1 is the superset given as R1:{ Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno} the Ssn attribute is the relation attribute that links the employee table to other tables. The superset satisfies the reflexivity rule since it shows direct functional dependency with all the subsets. Another set of attributes that shows reflexivity is the department FD R2: { Dname, Dnumber, Mgr_ssn, Mgr_start_date }. Attribute correspondence: FD. X

From the superset, we can now generate the R. Y subsets that have corresponding attributes. The relationship between the superset and the subset can therefore be written as $R1 < R5$ these two functional dependencies show attribute correspondence. For the case of R2 this can be represented as $R2 < R3$.

Transitivity: when X

In the other case $R3 < R4$ therefore $R2 < R4$. The above examples represent the inclusion dependencies found in the relation schema.