NORMALISATION OF TABLES:

1.) Insurance

Insurance_id -> company_name,start_date,end_date,co_insurance

Finding Closure:

(Insurance_ID)$^{+}$={ company_name,start_date,end_date,co_insurance, Insurance_id}

Candidate keys={Insurance_id}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

2.) Patient1

Patient_id->patient_id, polyclinic_name, patient_name,dob,insurance_id, sex, problem_or_disease, Dno,Doc_id, registration_time, registration_date

Patient_name->patient_id

Insurance_id->patient_id

Finding Closure:

(Patient_ID)$^{+}$={ Patient_ID , polyclinic_name Patient name, sex, problem_or_disease, Dno,Doc_id,registration_time, registration_date}

(Patient_name)$^{+}$={ Patient_ID , polyclinic_name Patient name, sex, problem_or_disease, Dno,Doc_id,registration_time, registration_date}

(Insurance_id)$^{+}$={ Patient_ID , polyclinic_name Patient name, sex, problem_or_disease, Dno,Doc_id,registration_time, registration_date}

Candidate keys={ Patient_ID, Patient_name, Insurance_id}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

3.) Patient_phone

Phone_number→patient_id

This table was already divided as I observed that the table Patient1 had multiple phone numbers associated with a single patient therefore the attribute phone_number which is multivalue attribute was divided and kept into different table.

Finding closure:

$(Phone\_number)^+$={ Phone_number,patient_id}

Candidate key={ Phone_number}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

4.) Visits:

Patient_id -> Patient name,registration_time,registration_date,final_details

Registration_time, ,registration_date -> Final_details

Patient_name -> patient_id

Finding Closure:

$(Patient\_ID)^+$={ Patient_ID ,Patient name,registration_time, registration_date, final_details}

$(Registration\_time, ,registration\_date)^+$= { registration_time, registration_date, final_details }

$(Patient\_name)^+$={ Patient_name,patient_id, registration_time, registration_date, final_details}

Candidate key={ Patient_ID ,Patient name }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Registration_time, ,registration_date being a non-prime attribute determines Final_details which is another non-prime attribute. Hence the table is not in 3NF. To convert it into 3NF, we make an entirely new table for

Registration_time, ,registration_date -> Final_details

Therefore we split table into two tables:

R_1 (Patient name,registration_time,registration_date)

R_2 (Registration_time, ,registration_date, final_details)

Next the table R_1 and R_2 are further in BCNF as the determining attributes are Super keys in R_1 and R_2

5.) Previous Visits:

Patient_id,visits -> prev_treatment taken from clinic

Finding Closure:

Patient_id,visits={prev_treatment taken from clinic,patient_id , visits}

Candidate key={ Patient_id,visits }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

6.) New patients

Patient_id → patient_name,insurance_id,discount_given

Patient_name → patient_id

Insurance_id → patient_id

Finding Closure:

$(Patient\_ID)^+$ ={ Patient_id, patient_name,insurance_id,discount_given}

$(Patient\_name)^+$={ Patient_id, patient_name,insurance_id,discount_given}

$(Insurance\_id)^+$= { Patient_id, patient_name,insurance_id,discount_given}

Candidate key={ Patient_ID, Patient_name, Insurance_id }

Normalization oftable:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

7.) Regular Visits

Patient_id → patient_name,insurance_id,discount_given

Patient_name → patient_id

Insurance_id → patient_id

Finding Closure:

$(Patient\_ID)^+$ ={ Patient_id, patient_name,insurance_id,discount_given}

$(Patient\_name)^+$={ Patient_id, patient_name,insurance_id,discount_given}

$(Insurance\_id)^+$= { Patient_id, patient_name,insurance_id,discount_given}

Candidate key={ Patient_ID, Patient_name, Insurance_id }

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

8.) DOCTORS

Doc_id → salary_slipno,doc_name,dep_no,dep_name

Salary_slipno→doc_id

Doc_name→doc_id

Dep_no→dep_name

Dep_name→dep_no

Finding closure:

$(Doc\_id)^+$ ={doc_id, salary_slipno,doc_name,dep_no,dep_name}

$(Salary\_slipno)^+$={ doc_id, salary_slipno,doc_name,dep_no,dep_name}

$(Doc\_name)^+$ = { doc_id, salary_slipno,doc_name,dep_no,dep_name}

$(Dep\_no)^+$ = { Dep_no,dep_name}

$(Dep\_name)^+$ = { Dep_no,dep_name}

Candidate key={ Doc_id, Salary_slipno, Doc_name}

Normalization of table:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Dep_no being a non-prime attribute determines dep_name which is another non-prime attribute. Hence the table is not in 3NF. To convert it into 3NF, we make an entirely new table for

Dep_no→dep_name

Therefore we split table into two tables:

R_1 (doc_id, salary_slipno,doc_name,dep_no)

R_2 (Dep_no,dep_name)

Next the table R_1 and R_2 are further in BCNF as the determining attributes are Super keys in R_1 and R_2

9.) DOCTOR_PHONE

Phone_number→doctor_id

This table was already divided as I observed that the table Doctors had multiple phone numbers associated with a single doctor therefore the attribute phone_number which is multivalue attribute was divided and kept into different table.

Finding closure:

$(Phone\_number)^+$={ Phone_number,doctor_id}

Candidate key={ Phone_number}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

10.) SALARY

Salary_slipno→salary,no_of_years_working

Finding closure:

$(Salary\_slipno)^+$={ Salary_slipno, salary,no_of_years_working }

Candidate key={ Salary_slipno }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

11.) DEPARTMENT:

Dep_no→ dep_name

Dep_name→dep_no

Finding closure:

$(Dep\_no)^+$={ Dep_no,dep_name}

$(Dep\_name)^+$={ Dep_no,dep_name}

Candidate key={ Dep_no,dep_name }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

12.) Endodontist

Doc_id -> rootcanal,charges

Root_canal -> doc_id

Finding closure:

$(Doc\_id)^+$ = {root_canal,charges,doc_id}

$(Root\_canal)^+$ = {root,canal,doc_id,charges }

Candidate key={ root_canal,doc_id }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

13.) Periodontist

Doc_id -> gums , price

Gums -> doc_id

Finding closure:

$(Doc\_id)^+$ = {gums , price,Doc_id }

$(Gums)^+$ = {doc_id,gums,price}

Candidate key={ doc_id,gums}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

14.) Gen_dentist

Doc_id ,cavities_or_missing teeth_or_mobile_teeth -> price

Finding closure:

$(Doc\_id$,cavities_or_missing teeth_or_mobile_teeth $)^+$ = { Doc_id ,cavities_or_missing teeth_or_mobile_teeth, price }

Candidate key={ Doc_id ,cavities_or_missing teeth_or_mobile_teeth }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

15.) Total_Bill

Patient_id ,bill_no -> insurance_id,patient_name,charges,discount_given,charges after discount,money_insurance , patient_pay,bill_no,cashier_id

Insurance_id -> Patient_id ,bill_no

Patient name -> Patient_id ,bill_no

Finding closure:

(Patient_id,bill _no)$^+$ =Patient_id,bill_no, insurance_id,patient_name, charges, discount_given, charges after discount,money_insurance , patient_pay,bill_no,cashier_id.

(Insurance_id) $^+$ = Patient_id, Bill_no, insurance_id, patient_name, charges, discount_given,charges after discount,money_insurance , patient_pay,bill_no,cashier_id.

(Patient_name) $^+$ = Patient_id, Bill_no, insurance_id, patient_name, charges,discount_given,charges after discount,money_insurance , patient_pay,bill_no,cashier_id.

Candidate key={ Patient_id ,bill_no, Insurance_id, Patient name }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

16.) Dependents

Depen name , patient_id -> phone number

Phone number -> patient_id,

dependent name,phone number -> patient_id

Finding closure:

(Depen_name ,patient_id)$^+$ = {Depen_name , patient_id, phone number}

(Phone number)$^+$ = {phone number ,patient_id}

(Dependent_name , phone number)$^+$ = {Dependent_name , phone number,patient_id}

Candidate key={ Dependent_name , phone number, patient_id}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

17.) Medical_History

Patient_id, past_treatment->allergies,pain_tooth,heart_probs,other_illness

Finding closure:

(Patient_id, past_treatment)$^{+}$={ allergies,pain_tooth,heart_probs,other_illness, Patient_id, past_treatment }

Candidate keys={ Patient_id, past_treatment }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

18.) CASHIER

Cashier_id->Name,Salary

Name->cashier_id

Finding closure:

(Cashier_id)$^{+}$={ Name,Salary, Cashier_id}

(Name)$^{+}$={ Name,Salary, Cashier_id}

Candidate keys={ Name,Salary, Cashier_id }

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.

19.) CASHIER_PHONE

Phone_number→cashier_id

This table was already divided as I observed that the table Doctors had multiple phone numbers associated with a single doctor therefore the attribute phone_number which is multivalue attribute was divided and kept into different table.

Finding closure:

$(Phone\_number)^+$={ Phone_number,cashier_id}

Candidate key={ Phone_number}

Normalisation:

Since all the attributes of the relation are atomic values and all the tuples have single values for the domain of their attributes, the table is in 1NF.

Since the table is in 1NF and there are no partial dependencies, it is also in 2NF.

Since the table is in 2NF and there are no transitive dependencies, it is also in 3NF.

Since the table is in 3NF and the determining attributes are Super keys, the table is also in BCNF.