**Functional Dependencies, Canonical cover & Normalization**

* **User** (userName, firstName, lastName, email, contactNumber, password)

**Candidate keys**:

userName, email.

userName ⊆ User and userName → User

email ⊆ User. email → User

We defined userName as primary key since email can be changed by a user.

**Functional Dependencies**:

F = {userName → firstName, userName → lastName, userName → email, userName → contactNumber, userName → password}

**Canonical Cover**:

* Step 1: Making right hand side (RHS) a single attribute.

RHS is already a single attribute in all FDs.

* Step 2: Having Left Hand Side in simple form.

userName → firstName (No left redundancy)

userName → lastName (No left redundancy)

userName → email (No left redundancy)

userName → contactNumber (No left redundancy)

userName → password (No left redundancy)

* Step 3: Remove redundant FDs.
* For: userName → firstName

Let G = F – {userName → firstName}

G = {userName → lastName, userName → email, userName → contactNumber, userName → password}

userName+G = {lastName, email, contactNumber, password}

Since firstName ∉ userName+G so userName → firstName is not redundant.

* For: userName → lastName

Let G = F – {userName → lastName}

G = {userName → firstName, userName → email, userName → contactNumber, userName → password}

userName+G = {firstName, email, contactNumber, password}

Since lastName ∉ userName+G so userName → lastName is not redundant.

* For: userName → email

Let G = F – {userName → email}

G = {userName → firstName, userName → lastName, userName → contactNumber, userName → password}

userName+G = {firstName, lastName, contactNumber, password}

Since email ∉ userName+G so userName → email is not redundant.

* For: userName → contactNumber

Let G = F – {userName → contactNumber}

G = {userName → firstName, userName → lastName, userName → email, userName → password}

userName+G = {firstName, lastName, email, password}

Since contactNumber ∉ userName+G so userName → contactNumber is not redundant.

* For: userName → password

Let G = F – {userName → password}

G = {userName → firstName, userName → lastName, userName → email, userName → contactNumber}

userName+G = {firstName, lastName, email, contactNumber}

Since password ∉ userName+G so userName → password is not redundant.

Since there is no redundant Functional Dependencies in F so F is a Canonical Cover of itself.

**Normalization**

The primary key of User is “userName”.

F = {userName → firstName, userName → lastName, userName → email, userName → contactNumber, userName → password}

Since LHS of all FDs in F is the primary key so User is in BCNF.

**Summary**

Primary Key: userName

Functional Dependencies: {userName → firstName, userName → lastName, userName → email, userName → contactNumber, userName → password}

Canonical Cover: {userName → firstName, userName → lastName, userName → email, userName → contactNumber, userName → password}

Normalization: BCNF

* **Employer** (userName, employerName, accStatus, category, balance)

**Candidate key**:

userName

userName ⊆ Employer and userName → Employer

**Functional Dependencies**:

F = {userName → employerName, userName → accStatus, username → category, userName → balance}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

{userName → employerName} (No left redundancy)

{userName → accStatus} (No left redundancy)

{userName → category} (No left redundancy)

{userName → balance} (No left redundancy)

* Step 3: Removing redundant FDs
* For: userName → employerName

Let G = F – {userName → employerName}

G = {userName → accStatus, userName → category, userName → balance}

userName+G = {accStatus, category, balance}

Since employerName ∉ userName+G so userName → employer is not redundant.

* For: userName → accStatus

Let G = F – {userName → accStatus}

G = {userName → employerName, userName → category, userName → balance}

userName+G = {employerName, category, balance}

Since accStatus ∉ userName+G so userName → accStatus is not redundant.

* For: userName → category

Let G = F – {userName → category}

G = {userName → employerName, userName → accStatus, userName → balance}

userName+G = {employerName, accStatus, balance}

Since category ∉ userName+G so userName → category is not redundant.

* For: userName → balance

Let G = F – {userName → balance}

G = {userName → employerName, userName → accStatus}

userName+G = {employerName, accStatus}

Since balance ∉ userName+G so userName → balance is not redundant.

There are no redundant functional dependencies in F. So F is a canonical cover of itself.

**Normalization**

Primary Key: userName

F = {userName → employerName, userName → accStatus, userName → category, userName → balance}

Since LHS of all FD’s in F is the primary key so Employer is in BCNF.

**Summary**

Primary Key: userName

Functional Dependencies: {userName → employerName, userName → accStatus, userName → category, userName → balance}

Canonical Cover: {userName → employerName, userName → accStatus, userName → category, userName → balance}

Normalization: BCNF

* **Applicant** (userName, category, accStatus)

**Candidate key**:

userName

userName ⊆ Applicant and userName → Applicant

**Functional Dependencies**:

F = {userName → category, userName → accStatus}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

{userName → accStatus} (No left redundancy)

* Step 3: Removing redundant FDs
* For: userName → category

Let G = F – {userName → category}

G = {userName → accStatus}

userName+G = {accStatus}

Since category ∉ userName+G so userName → category is not redundant.

* For: userName → accStatus

Let G = F – {userName → accStatus}

G = {userName → category}

userName+G = {category}

Since accStatus ∉ userName+G so userName → accStatus is not redundant

There are no redundant functional dependencies in F. So F is a canonical cover of itself.

**Normalization**

Primary Key: userName

F = {userName → category , userName → accStatus}

Since LHS of all FD’s in F is the primary key so Applicant is in BCNF.

**Summary**

Primary Key: userName

Functional Dependencies: {userName → category, userName → accStatus}

Canonical Cover: {userName → category, userName → accStatus}

Normalization: BCNF

* **Admin** (userName)

**Candidate Key:** userName.

userName ⊆ Admin and userName → Admin

**Functional Dependencies:** F = {}

**Canonical Cover:** F is a canonical cover of itself.

**Normalization:** BCNF

**Summary**

Primary Key: userName

Functional Dependencies: {}

Canonical Cover: {}

Normalization: BCNF

* **Job** (jobID, employerUserName, title, datePosted, description, category, jobStatus, empNeeded)

**Candidate key**:

jobID

jobID ⊆ Job and jobID → Job

**Functional Dependencies**:

F = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

{jobID → employerUserName} (No left redundancy)

{jobID → title} (No left redundancy)

jobID → datePosted} (No left redundancy)

{jobID → description} (No left redundancy)

{jobID → category} (No left redundancy)

{jobID → jobStatus} (No left redundancy)

{jobID → empNeeded} (No left redundancy)

* Step 3: Removing redundant FDs
* For: jobID → employerUserName

Let G = F – {jobID → employerUserName}

G = {jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

jobID+G = {title, datePosted, description, category, jobStatus, empNeeded}

Since employerUserName∉ jobID+G so jobID → employerUserNameis not redundant.

* For: jobID → title

Let G = F – {jobID → title}

G = {jobID → employerUserName, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

jobID+G = {employerUserName, datePosted, description, category, jobStatus, empNeeded}

Since title∉ jobID+G so jobID → titleis not redundant.

* For: jobID → datePosted

Let G = F – {jobID → datePosted}

G = {jobID → employerUserName, jobID → title, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

jobID+G = {employerUserName, title, description, category, jobStatus, empNeeded}

Since datePosted ∉ jobID+G so jobID → datePosted is not redundant.

* For: jobID → description

Let G = F – {jobID → description}

G = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → category, jobID → jobStatus, jobID → empNeeded}

jobID+G = {employerUserName, title, datePosted, category, jobStatus, empNeeded}

Since description ∉ jobID+G so jobID → description is not redundant.

* For: jobID → category

Let G = F – {jobID → category}

G = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → jobStatus, jobID → empNeeded}

jobID+G = {employerUserName, title, datePosted, description, jobStatus, empNeeded}

Since category ∉ jobID+G so jobID → category is not redundant.

* For: jobID → jobStatus

Let G = F – {jobID → jobStatus}

G = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → empNeeded}

jobID+G = {employerUserName, title, datePosted, description, category, empNeeded}

Since jobStatus ∉ jobID+G so jobID → jobStatus is not redundant.

* For: jobID → empNeeded

Let G = F – {jobID → empNeeded}

G = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus}

jobID+G = {employerUserName, title, datePosted, description, category, jobStatus}

Since empNeeded ∉ jobID+G so jobID → empNeeded is not redundant.

There are no redundant functional dependencies in F. So F is a canonical cover of itself.

**Normalization**

Primary Key: jobID

F = {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

Since LHS of all FD’s in F is the primary key so Job is in BCNF.

**Summary**

Primary Key: jobID

Functional Dependencies: {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

Canonical Cover: {jobID → employerUserName, jobID → title, jobID → datePosted, jobID → description, jobID → category, jobID → jobStatus, jobID → empNeeded}

Normalization: BCNF

* **CreditCardInfo** (CCNumber, expireDate, userUserName, CCBNumber, defaultCard, auto\_manual)

**Candidate key**:

{CCNumber, expireDate}

{CCNumber, expireDate} ⊆ CreditCardInfo and {CCNumber, expireDate} → CreditCardInfo

**Functional Dependencies**:

F = {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber, expireDate → defaultCard, expireDate → auto\_manual}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

CCNumber+ = CCNumber.

expireDate+ = expireDate.

CCNumber, expireDate → userUserName (No left redundancy)

CCNumber, expireDate → CCBNumber (No left redundancy)

CCNumber, expireDate → defaultCard (No left redundancy)

CCNumber, expireDate → auto\_manual (No left redundancy)

* Step 3: Removing redundant FDs
* For: CCNumber, expireDate → userUserName

Let G = F – {CCNumber, expireDate → userUserName}

G = {CCNumber, expireDate → CCBNumber, CCNumber, expireDate → defaultCard, CCNumber, expireDate → auto\_manual}

{CCNumber, expireDate}+G = {CCBNumber, defaultCard, auto\_manual}

Since userUserName ∉ {CCNumber, expireDate}+G so CCNumber, expireDate → userUserName is not redundant.

* For: CCNumber, expireDate → CCBNumber

Let G = F – {CCNumber, expireDate → CCBNumber}

G = {CCNumber, expireDate → userUserName, CCNumber, expireDate → defaultCard, CCNumber, expireDate → auto\_manual}

{CCNumber, expireDate}+G = {userUserName, defaultCard, auto\_manual}

Since CCBNumber ∉ {CCNumber, expireDate}+G so CCNumber, expireDate → CCBNumber is not redundant.

* For: CCNumber, expireDate → defaultCard

Let G = F – {CCNumber, expireDate → defaultCard}

G = {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber, CCNumber, expireDate → auto\_manual}

{CCNumber, expireDate}+G = {userUserName, CCBNumber, auto\_manual}

Since defaultCard ∉ {CCNumber, expireDate}+G so CCNumber, expireDate → defaultCard is not redundant.

* For: CCNumber, expireDate → auto\_manual

Let G = F – {CCNumber, expireDate → auto\_manual}

G = {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber, CCNumber, expireDate → defaultCard}

{CCNumber, expireDate}+G = {userUserName, CCBNumber, defaultCard}

Since auto\_manual ∉ {CCNumber, expireDate}+G so CCNumber, expireDate → auto\_manual is not redundant.

There are no redundant functional dependencies in F. So, F is a canonical cover of itself.

**Normalization**

Primary Key: CCNumber, expireDate

F = {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber, expireDate → defaultCard, expireDate → auto\_manual}

Since LHS of all FD’s in F is the primary key so CreditCardInfo is in BCNF.

**Summary**

Primary Key: CCNumber, expireDate

Functional Dependencies: {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber}

Canonical Cover: {CCNumber, expireDate → userUserName, CCNumber, expireDate → CCBNumber}

Normalization: BCNF

* **PADInfo** (accountNumber, instituteNumber, branchNumber, defaultAccount, auto\_manual)

**Candidate key**:

accountNumber

accountNumber ⊆ PADInfo and accountNumber → PADInfo

**Functional Dependencies**:

F = {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

{accountNumber → instituteNumber} (No left redundancy)

{accountNumber → branchNumber} (No left redundancy)

{accountNumber → defaultAccount} (No left redundancy)

{accountNumber → auto\_manual} (No left redundancy)

* Step 3: Removing redundant FDs
* For: accountNumber → instituteNumber

Let G = F – {accountNumber → instituteNumber}

G = {accountNumber → userUserName, accountNumber → branchNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

accountNumber+G = {branchNumber, defaultAccount, auto\_manual}

Since instituteNumber ∉ accountNumber+G so accountNumber → instituteNumber is not redundant.

* For: accountNumber → branchNumber

Let G = F – {accountNumber → branchNumber}

G = accountNumber → instituteNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

accountNumber+G = {instituteNumber, defaultAccount, auto\_manual}

Since branchNumber ∉ accountNumber+G so accountNumber → branchNumber is not redundant.

* For: accountNumber → defaultAccount

Let G = F – {accountNumber → defaultAccount }

G = {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → auto\_manual}

accountNumber+G = {instituteNumber, brachNumber, auto\_manual}

Since defaultAccount ∉ accountNumber+G so accountNumber → defaultAccount is not redundant.

* For: accountNumber → auto\_manual

Let G = F – {accountNumber → auto\_manual}

G = {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → defaultAccount}

accountNumber+G = {instituteNumber, brachNumber, defaultAccount}

Since auto\_manual ∉ accountNumber+G so accountNumber → auto\_manual is not redundant.

There are no redundant FD. So, F is a canonical cover of itself.

**Normalization**

Primary Key: accountNumber

F = {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

Since LHS of all FD’s in F is the primary key so PADInfo is in BCNF.

**Summary**

Primary Key: accountNumber

Functional Dependencies: {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

Canonical Cover: {accountNumber → instituteNumber, accountNumber → branchNumber, accountNumber → defaultAccount, accountNumber → auto\_manual}

Normalization: BCNF

* **ApplicantBalane** (applicantUserName, balance)

**Candidate key**:

applicantUserName

applicantUserName ⊆ ApplicantBalane and applicantUserName → ApplicantBalane

**Functional Dependencies**:

F = {applicantUserName → balance}

**Canonical Cover:**

There are no redundant FD. So, F is a canonical cover of itself.

**Normalization**

Primary Key: PADAccountNumber

F = {PADAccountNumber → branchNumber}

Since LHS of all FD’s in F is the primary key so PADBranch is in BCNF.

**Summary**

Primary Key: PADAccountNumber

Functional Dependencies: {PADAccountNumber → branchNumber}

Canonical Cover: {PADAccountNumber → branchNumber}

Normalization: BCNF

* **Application** (applicantUserName, jobID, applicationStatus, applicationDate)

**Candidate key**:

{applicantUserName, jobID}

{applicantUserName, jobID} ⊆ Apply and {applicantUserName, jobID} → Apply

**Functional Dependencies**:

F = {applicantUserName, jobID → applicationStatus, applicantUserName, jobID → applicationDate}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

applicantUserName + = applicantUserName.

jobID+ = jobID.

{applicantUserName, jobID} → applicationStatus (No left redundancy)

{applicantUserName, jobID} → applicationDate (No left redundancy)

* Step 3: Removing redundant FDs
* For: {applicantUserName, jobID} → applicationStatus

Let G = F – {applicantUserName, jobID → applicationStatus}

G = {applicantUserName, jobID → applicationDate}

{applicantUserName, jobID}+G = {applicationDate}

Since applicationStatus ∉ {applicantUserName, jobID}+G so applicantUserName, jobID → applicationStatus is not redundant.

* For: {applicantUserName, jobID} → applicationDate

Let G = F – {applicantUserName, jobID → applicationDate}

G = {applicantUserName, jobID → applicationStatus}

{applicantUserName, jobID}+G = {applicationStatus}

Since applicationDate ∉ {applicantUserName, jobID}+G so applicantUserName, jobID → applicationDate is not redundant.

There are no redundant functional dependencies in F. So, F is a canonical cover of itself.

**Normalization**

Primary Key: applicantUserName, jobID

F = {applicantUserName, jobID → applicationStatus, applicantUserName, jobID → applicationDate}

Since LHS of all FD’s in F is the primary key so Apply is in BCNF.

**Summary**

Primary Key: applicantUserName, jobID

Functional Dependencies: {applicantUserName, jobID → applicationStatus, applicantUserName, jobID → applicationDate}

Canonical Cover: {applicantUserName, jobID → applicationStatus, applicantUserName, jobID → applicationDate}

Normalization: BCNF

* **EmployerCC** (employerUserName**,** CCNumber**)**

**Candidate key**:

{CCNumber}

{CCNumber} ⊆ EmployerCCand {CCNumber} → EmployerCC

**Functional Dependencies**:

F = {CCNumber → employerUserName}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

LHS of all FDs are in simple form.

* Step 3: Removing redundant FDs
* For: {CCNumber → employerUserName}

Let G = F – {CCNumber → employerUserName}

G = {}

Since employerUserName ∉ {CCNumber}+G so CCNumber → employerUserName is not redundant.

**Normalization**

Primary Key: CCNumber

F = {CCNumber → employerUserName}

Since LHS of all FD’s in F is the primary key so Apply is in BCNF.

**Summary**

Primary Key: CCNumber

Functional Dependencies: {CCNumber → employerUserName}

Canonical Cover: {CCNumber → employerUserName}

Normalization: BCNF

* **EmployerPAD** (employerUserName**,** accountNumber**)**

**Candidate key**:

{accountNumber}

{accountNumber} ⊆ EmployerPADand {accountNumber} → EmployerPAD

**Functional Dependencies**:

F = {accountNumber → employerUserName}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

LHS of all FDs are in simple form.

* Step 3: Removing redundant FDs
* For: {accountNumber → employerUserName}

Let G = F – {accountNumber → employerUserName}

G = {}

Since employerUserName ∉ {accountNumber}+G so accountNumber → employerUserName is not redundant.

**Normalization**

Primary Key: accountNumber

F = {accountNumber → employerUserName}

Since LHS of all FD’s in F is the primary key so Apply is in BCNF.

**Summary**

Primary Key: accountNumber

Functional Dependencies: {accountNumber → employerUserName}

Canonical Cover: {accountNumber → employerUserName}

Normalization: BCNF

* **ApplicantCC** (applicantUserName**,** CCNumber**)**

**Candidate key**:

{CCNumber}

{CCNumber} ⊆ ApplicantCCand {CCNumber} → ApplicantCC

**Functional Dependencies**:

F = {CCNumber → applicantUserName}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

LHS of all FDs are in simple form.

* Step 3: Removing redundant FDs
* For: {CCNumber → applicantUserName}

Let G = F – {CCNumber → applicantUserName}

G = {}

Since applicantUserName ∉ {CCNumber}+G so CCNumber → applicantUserName is not redundant.

**Normalization**

Primary Key: CCNumber

F = {CCNumber → applicantUserName}

Since LHS of all FD’s in F is the primary key so Apply is in BCNF.

**Summary**

Primary Key: CCNumber

Functional Dependencies: {CCNumber → applicantUserName}

Canonical Cover: {CCNumber → applicantUserName}

Normalization: BCNF

* **EmployerPAD** (applicantUserName**,** accountNumber**)**

**Candidate key**:

{accountNumber}

{accountNumber} ⊆ ApplicantPADand {accountNumber} → ApplicantPAD

**Functional Dependencies**:

F = {accountNumber → applicantUserName}

**Canonical Cover:**

* Step 1: Making RHS single attribute.

All the FDs have single attribute in the RHS.

* Step 2: Having LHS in simple form.

LHS of all FDs are in simple form.

* Step 3: Removing redundant FDs
* For: {accountNumber → applicantUserName}

Let G = F – {accountNumber → applicantUserName}

G = {}

Since applicantUserName ∉ {accountNumber}+G so accountNumber → applicantUserName is not redundant.

**Normalization**

Primary Key: accountNumber

F = {accountNumber → applicantUserName}

Since LHS of all FD’s in F is the primary key so Apply is in BCNF.

**Summary**

Primary Key: accountNumber

Functional Dependencies: {accountNumber → applicantUserName}

Canonical Cover: {accountNumber → applicantUserName}

Normalization: BCNF