

COL362 Project – Milestone 1

Database design

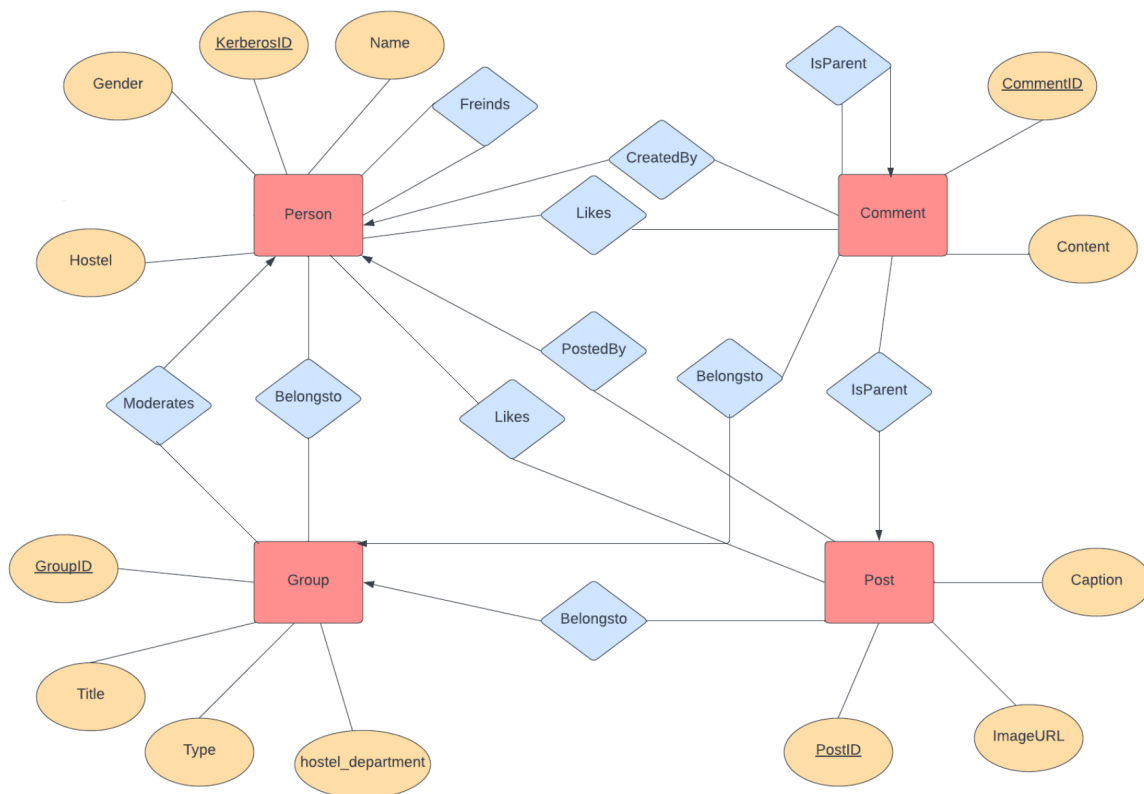
Power-saver-mode.h

Harshit Mawandia (2020CS10348)

Rajat Bhardwaj (2020CS50436)

Rishita Agrawal (2020CS50439)

ER Diagram



Functional dependencies

The functional Dependencies of each relation are as follows

1. Person

Kerberos_ID -> Name

Kerberos_ID -> Hostel

Kerberos_ID -> Gender

2. Person_Belongsto_Group

(PersonID, GroupID) -> PersonID

(PersonID, GroupID) -> GroupID

3. Post

PostID -> Image_URL

PostID -> Caption

PostID -> CreatorPersonID

PostID -> BelongsTo_Group

4. Comments

CommentID -> Content

CommentID -> CreatorPersonID

CommentID -> ParentPostID

CommentID -> ParentCommentID

5. Person_Likes_Post

(PersonID, PostID) -> PersonID

(PersonID, PostID) -> PostID

6. Person_Likes_Comment

(PersonID, CommentID) -> PersonID

(PersonID, CommentID) -> CommentID

7. Friends

(Person1ID, Person2ID) -> Person1ID

(Person1ID, Person2ID) -> Person2ID

8. Groups

GroupID -> Type

GroupID -> Hostel_Department

GroupID -> Title

GroupID -> Moderator

FD preserving normalization

We will show that all the relations satisfy Boyce-Codd Normal Form

First Normal Form

Every tuple must contain an atomic value

Second Normal Form

It should satisfy the first normal form.

There should not be any partial dependencies. No non-prime attribute is functionally dependent on a proper subset of any candidate key.

Third Normal Form

It should satisfy the second normal form.

No transitive dependencies. Any attribute can not depend on a non-prime attribute (that is not a part of the primary key)

Boyce Codd Normal Form

It should satisfy the third normal form.

For all $X \rightarrow Y$, X should be a super key.

1. Person(Kerberos_ID, Name, Hostel, Gender) : Candidate Key / Primary Key- Kerberos_ID

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : Since there does-not exist any proper subset of candidate key(since it only has 1 attribute)
- Third Normal form : No combination of non-prime attributes can uniquely determine any other attribute
- Boyce Codd Normal Form : Since primary key (Kerberos_ID) is itself a super key(contains minimal attributes) therefore it satisfies Boyce Codd Normal Form

2. Person_Belongsto_Group(PersonID, GroupID) : Candidate Key / Primary Key - (PersonID, GroupID)

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : No partial subset of the Primary key can determine the tuple
- Third Normal form : There does not exist any attributes that can combine to form a non-primary key
- Boyce Codd Normal Form : Trivial proof

3. Post(PostID, Image_URL, Caption, CreatorPersonID, BelongsTo_Group) : Candidate key / Primary key - PostID

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : Since there does-not exist any proper subset of candidate key(since it only has 1 attribute)
- Third Normal form : No combination of non-prime attributes can uniquely determine any other attribute
- Boyce Codd Normal Form : Since primary key (PostID) is itself a super key(contains minimal attributes) therefore it satisfies Boyce Codd Normal Form

4. Comments(CommentID, Content, CreatorPersonID, ParentPostID, ParentCommentID) : Candidate key / Primary key - CommentID

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : Since there does-not exist any proper subset of candidate key(since it only has 1 attribute)
- Third Normal form : No combination of non-prime attributes can uniquely determine any other attribute
- Boyce Codd Normal Form : Since primary key (CommentID) is itself a super key(contains minimal attributes) therefore it satisfies Boyce Codd Normal Form

5. Person_Likes_Post(PersonID, PostID) : Candidate key / primary key - (PersonID, PostID)

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : No partial subset of the Primary key can determine the tuple
- Third Normal form : There does not exist any attributes that can combine to form a non-primary key

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : No partial subset of the Primary key can determine the tuple
- Third Normal form : There does not exist any attributes that can combine to form a non-primary key
- Boyce Codd Normal Form : Trivial proof

7. Friends(Person1ID, Person2ID) : Candidate key / primary key - (Person1ID, Person2ID)

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : No partial subset of the Primary key can determine the tuple
- Third Normal form : There does not exist any attributes that can combine to form a non-primary key
- Boyce Codd Normal Form : Trivial proof

- First Normal form : Since all attributes have atomic value, it satisfies the first normal form
- Second Normal form : Since there does-not exist any proper subset of candidate key(since it only has 1 attribute)
- Third Normal form : No combination of non-prime attributes can uniquely determine any other attribute
- Boyce Codd Normal Form : Since primary key (GroupID) is itself a super key(contains minimal attributes) therefore it satisfies Boyce Codd Normal Form

Relational Schema



Github repository for the project

We have added you to the repository “power-saver-mode.h”