University-Research-Conference-DBMS

Database Design | ERD | Normalization (UNF to 3NF)

By Abrar Shakeel Ghoury

Project Type: Academic Database Modeling

Tech Used: Draw.io, SQL, DBMS,MS Excel

Date: JULY 2025

Abstract

This project presents a conceptual and logical design of a **University Research & Conference Management System**, aimed at tracking academic research activities, student and faculty contributions, publications, and conference participation. Using **Entity Relationship Diagram (ERD)** and **normalization techniques up to 3rd Normal Form (3NF)**, this system captures essential entities such as students, faculty, research projects, publications, and conferences. It addresses real-world academic relationships like authorship order, research involvement hours, and role-based participation in events. The final schema ensures data integrity, avoids redundancy, and supports future implementation in both **relational and NoSQL environments**. This project is a demonstration of data modeling principles applied to a realistic academic scenario.

Table of Contents

Problem Statement

Modern universities are increasingly focused on improving their academic visibility through **faculty-led research projects**, **student involvement in innovation**, and **active participation in national and international conferences**. However, managing this complex ecosystem using spreadsheets or disconnected tools often leads to data redundancy, inconsistency, and inefficiency.

To address this, a comprehensive **Research & Conference Management System** is required that can track:

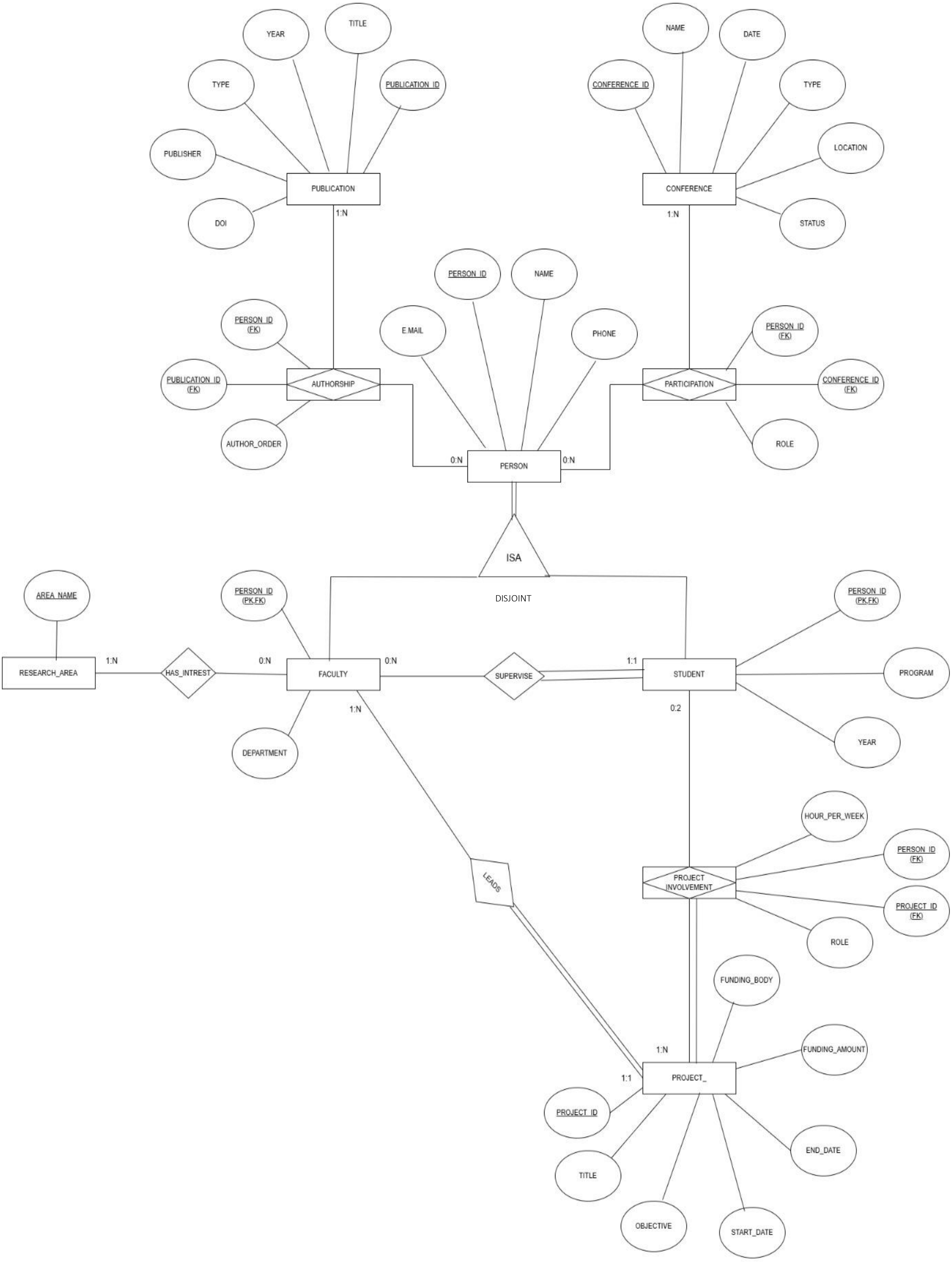
- Research projects, their objectives, timelines, and funding bodies
- Student and faculty participation with specific roles and time commitments
- Authorship information and publication metadata (with author order)
- Conference details along with participant roles (e.g., presenter, organizer, keynote speaker)

The system must also support academic constraints such as:

- A student participating in no more than two active projects
- Publications always having at least one faculty member as an author
- Conferences marked active only when participants are linked

The goal of this project is to **design a normalized relational database model**, starting from raw unstructured data (UNF) and progressing to **3rd Normal Form (3NF)**, backed by a well-defined **Entity-Relationship Diagram (ERD)**. This ensures minimal redundancy, referential integrity, and readiness for future implementation in both relational and NoSQL environments.

Conceptual Schema – Entity Relationship Diagram (ERD)



Business Rules and Constraint

The following real-world constraints and business rules are applicable to the system and must be maintained through application logic, triggers, or relational design:

1. A student cannot be involved in more than two active projects at the same time.
2. Every publication must have at least one faculty author.
3. Conference cannot be marked as “Active” unless at least one participant is linked to it.
4. A person can only be either a student or a faculty (disjoint specialization).
5. Multiple roles per person per project or conference are allowed.
6. Authorship must preserve the author order for academic recognition.
7. Funding for a project can come from multiple bodies with separate amounts.

ERD Explanation

his project models a university's research and conference activities through a well-structured Entity Relationship Diagram (ERD). At the center is the **Person** entity, which is specialized into **Student** and **Faculty** using a disjoint ISA relationship. This ensures that one person can only belong to one category.

Each person can have multiple phone numbers and research areas, handled through **Person_Phone** and **Person_Area** tables to maintain atomic data and support normalization.

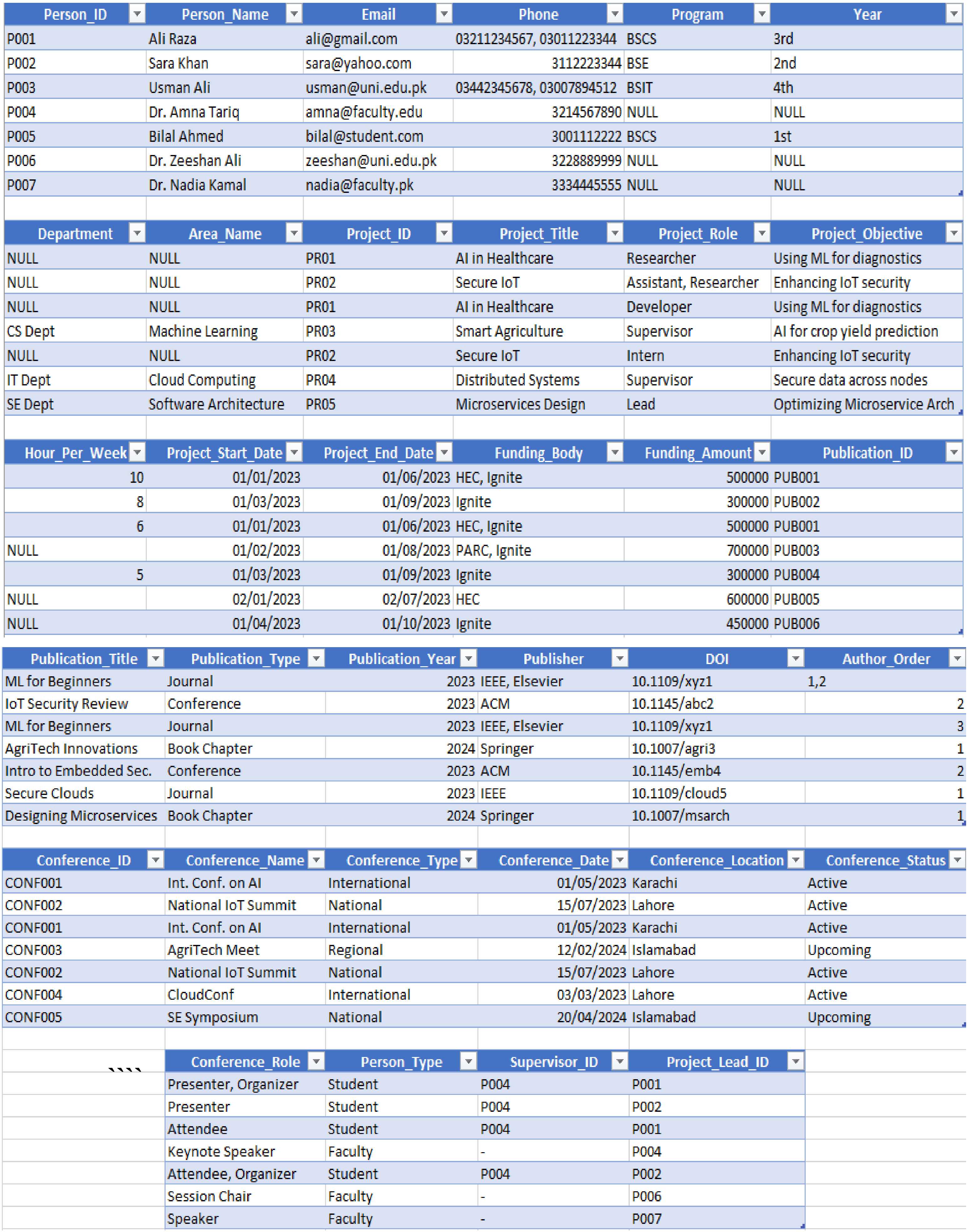
Students and faculty can contribute to multiple research projects through the **Person_Project_Role** table, which stores their specific roles such as Researcher, Developer, or Supervisor. Faculty members may also lead projects **(Leads_Project)** and supervise students **(Supervise),** reflecting real-world academic relationships.

Publications can have multiple authors, and the correct author sequence is maintained using the **Authorship** table. A single publication may also be associated with more than one publisher, which is managed using the **Publication_Publisher** table.

Conference participation is modeled through the **Conference_Participation** table, where individuals may take part in different roles such as Presenter, Organizer, or Attendee.

The ERD effectively resolves all many-to-many and multi-valued relationships, supports database normalization, and provides a solid foundation for designing a scalable, relational academic data system.

UNF Table – Un-Normalized Form



Person_ID	Person_Name	Email	Phone	Program	Year
P001	Ali Raza	ali@gmail.com	03211234567, 03011223344	BSCS	3rd
P002	Sara Khan	sara@yahoo.com	3112223344	BSE	2nd
P003	Usman Ali	usman@uni.edu.pk	03442345678, 03007894512	BSIT	4th
P004	Dr. Amna Tariq	amna@faculty.edu	3214567890	NULL	NULL
P005	Bilal Ahmed	bilal@student.com	3001112222	BSCS	1st
P006	Dr. Zeeshan Ali	zeeshan@uni.edu.pk	3228889999	NULL	NULL
P007	Dr. Nadia Kamal	nadia@faculty.pk	3334445555	NULL	NULL

Department	Area_Name	Project_ID	Project_Title	Project_Role	Project_Objective
NULL	NULL	PR01	AI in Healthcare	Researcher	Using ML for diagnostics
NULL	NULL	PR02	Secure IoT	Assistant, Researcher	Enhancing IoT security
NULL	NULL	PR01	AI in Healthcare	Developer	Using ML for diagnostics
CS Dept	Machine Learning	PR03	Smart Agriculture	Supervisor	AI for crop yield prediction
NULL	NULL	PR02	Secure IoT	Intern	Enhancing IoT security
IT Dept	Cloud Computing	PR04	Distributed Systems	Supervisor	Secure data across nodes
SE Dept	Software Architecture	PR05	Microservices Design	Lead	Optimizing Microservice Arch

Hour_Per_Week	Project_Start_Date	Project_End_Date	Funding_Body	Funding_Amount	Publication_ID
10	01/01/2023	01/06/2023	HEC, Ignite	500000	PUB001
8	01/03/2023	01/09/2023	Ignite	300000	PUB002
6	01/01/2023	01/06/2023	HEC, Ignite	500000	PUB001
NULL	01/02/2023	01/08/2023	PARC, Ignite	700000	PUB003
5	01/03/2023	01/09/2023	Ignite	300000	PUB004
NULL	02/01/2023	02/07/2023	HEC	600000	PUB005
NULL	01/04/2023	01/10/2023	Ignite	450000	PUB006

Publication_Title	Publication_Type	Publication_Year	Publisher	DOI	Author_Order
ML for Beginners	Journal	2023	IEEE, Elsevier	10.1109/xyz1	1,2
IoT Security Review	Conference	2023	ACM	10.1145/abc2	2
ML for Beginners	Journal	2023	IEEE, Elsevier	10.1109/xyz1	3
AgriTech Innovations	Book Chapter	2024	Springer	10.1007/agri3	1
Intro to Embedded Sec.	Conference	2023	ACM	10.1145/emb4	2
Secure Clouds	Journal	2023	IEEE	10.1109/cloud5	1
Designing Microservices	Book Chapter	2024	Springer	10.1007/msarch	1

Conference_ID	Conference_Name	Conference_Type	Conference_Date	Conference_Location	Conference_Status
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF003	AgriTech Meet	Regional	12/02/2024	Islamabad	Upcoming
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF004	CloudConf	International	03/03/2023	Lahore	Active
CONF005	SE Symposium	National	20/04/2024	Islamabad	Upcoming

Conference_Role	Person_Type	Supervisor_ID	Project_Lead_ID
Presenter, Organizer	Student	P004	P001
Presenter	Student	P004	P002
Attendee	Student	P004	P001
Keynote Speaker	Faculty	-	P004
Attendee, Organizer	Student	P004	P002
Session Chair	Faculty	-	P006
Speaker	Faculty	-	P007

Anomalies in UNF

The UNF table contains raw, merged data with multi-valued and repeating fields. As a result, it introduces the following data anomalies:

1. Insertion Anomaly:

New data (e.g., a new research area or phone number) cannot be added unless we repeat other unrelated data like project or publication.

2. Deletion Anomaly:

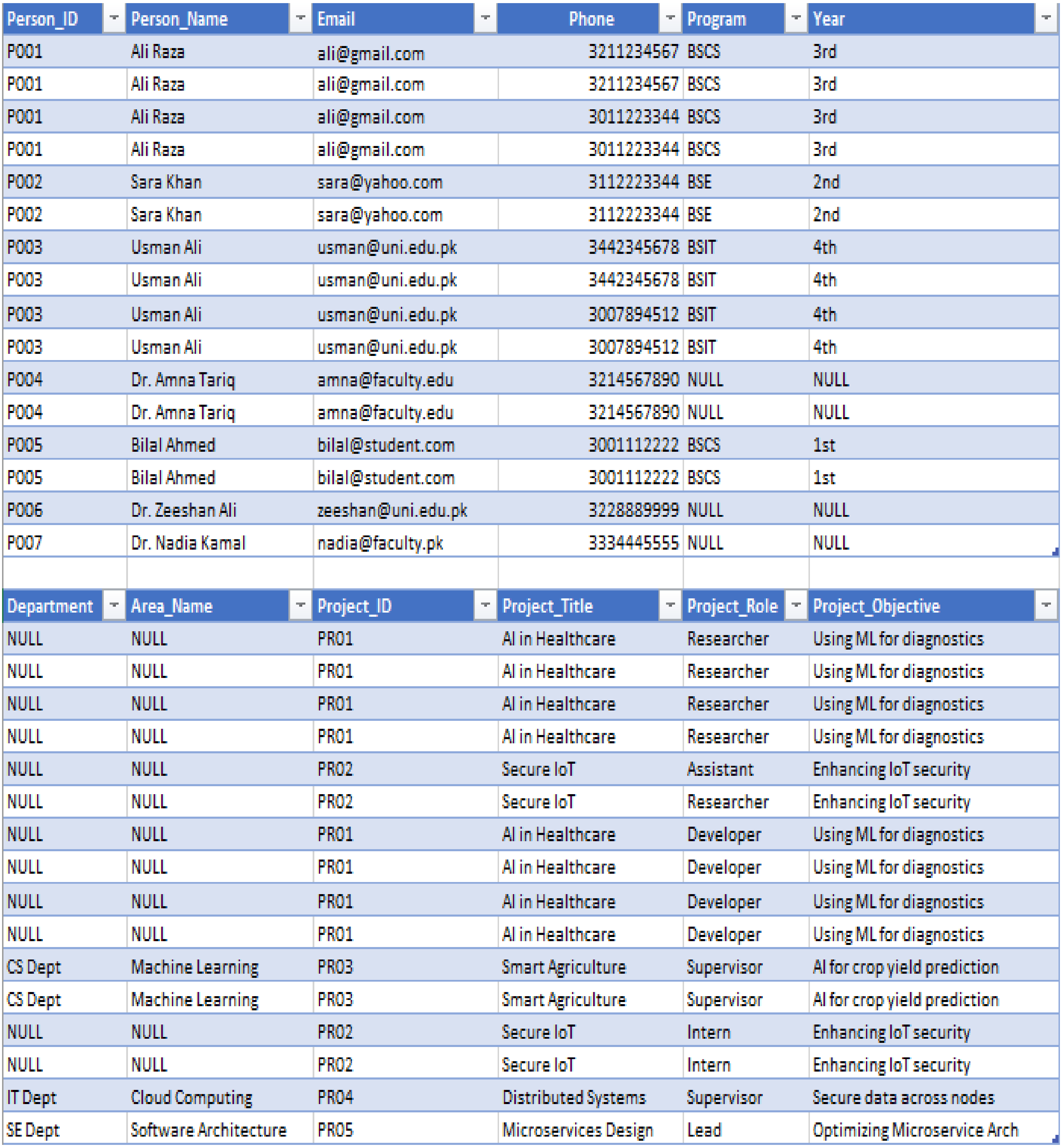
If we delete a row where a person's project ends, we may lose their publication or phone record as well — even if it is still valid.

3. Update Anomaly:

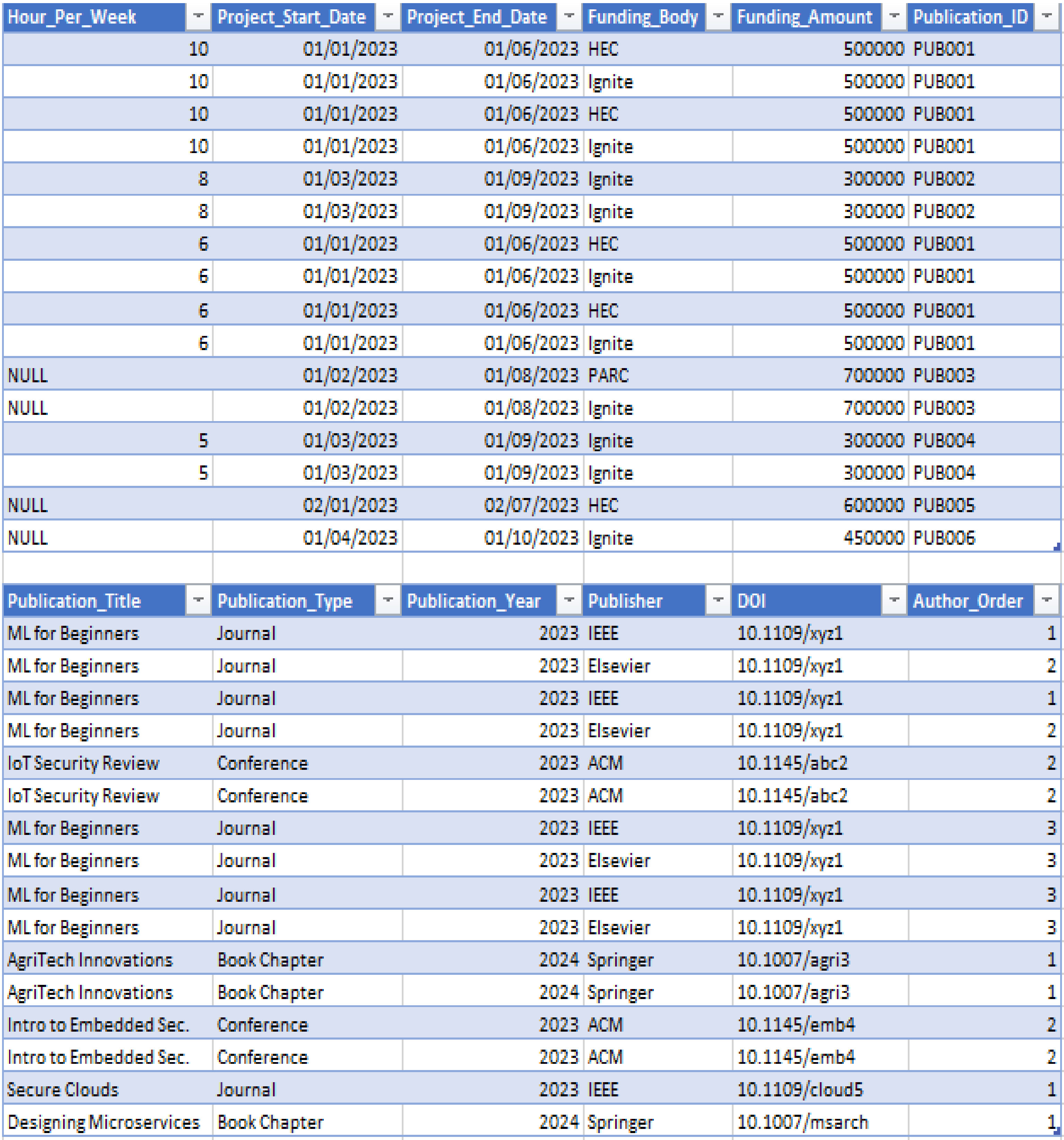
If a person's name or email is stored in multiple rows and one value changes, we must manually update it everywhere to avoid inconsistency.

Logical Schema:

First Normal Form (1-NF) – Table Structure

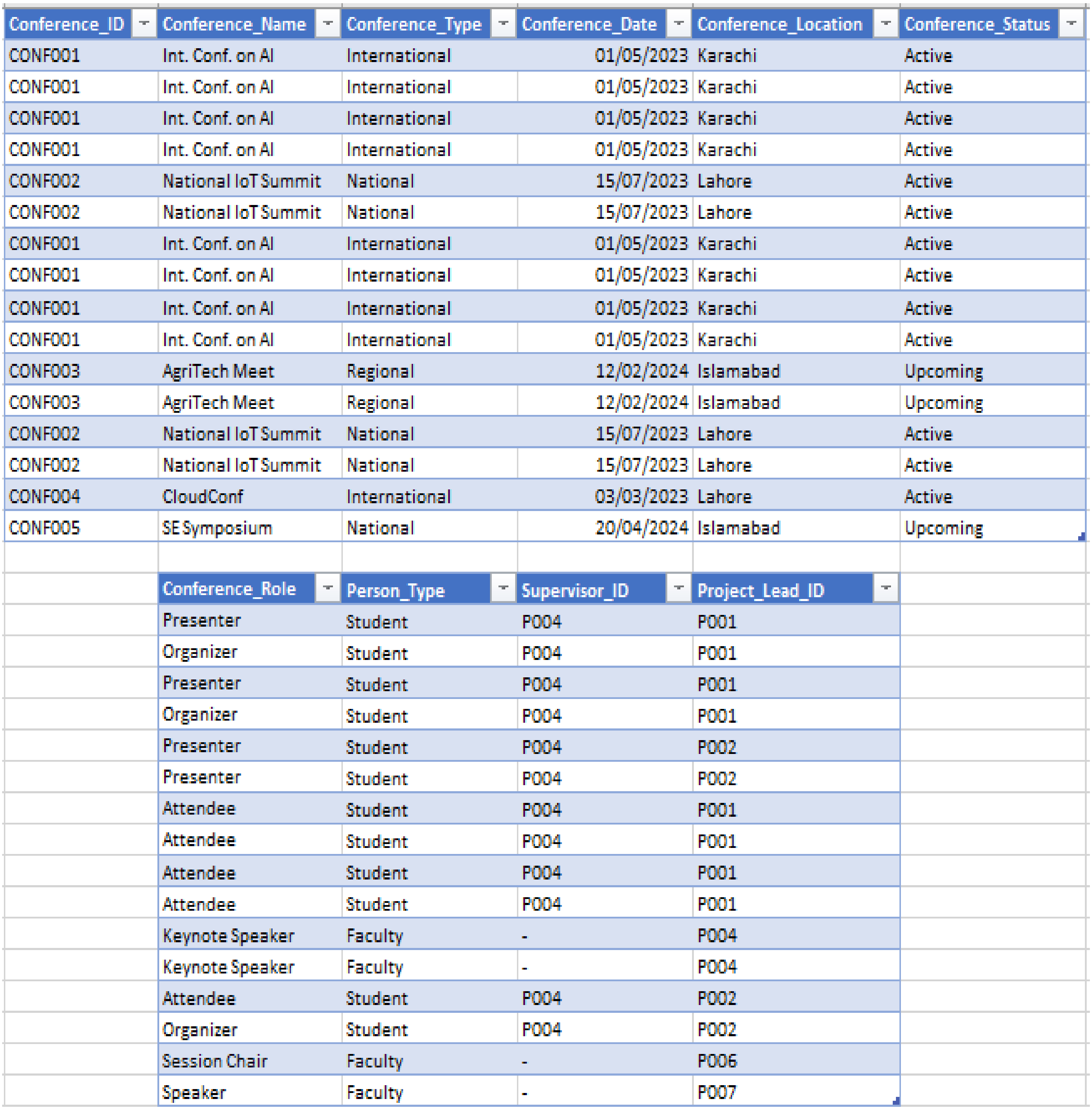


Person_ID	Person_Name	Email	Phone	Program	Year	Department	Area_Name	Project_ID	Project_Title	Project_Role	Project_Objective
P001	Ali Raza	ali@gmail.com	3211234567	BSCS	3rd	NULL	NULL	PR01	AI in Healthcare	Researcher	Using ML for diagnostics
P001	Ali Raza	ali@gmail.com	3211234567	BSCS	3rd	NULL	NULL	PR01	AI in Healthcare	Researcher	Using ML for diagnostics
P001	Ali Raza	ali@gmail.com	3011223344	BSCS	3rd	NULL	NULL	PR01	AI in Healthcare	Researcher	Using ML for diagnostics
P001	Ali Raza	ali@gmail.com	3011223344	BSCS	3rd	NULL	NULL	PR01	AI in Healthcare	Researcher	Using ML for diagnostics
P002	Sara Khan	sara@yahoo.com	3112223344	BSE	2nd	NULL	NULL	PR02	Secure IoT	Assistant	Enhancing IoT security
P002	Sara Khan	sara@yahoo.com	3112223344	BSE	2nd	NULL	NULL	PR02	Secure IoT	Researcher	Enhancing IoT security
P003	Usman Ali	usman@uni.edu.pk	3442345678	BSIT	4th	NULL	NULL	PR01	AI in Healthcare	Developer	Using ML for diagnostics
P003	Usman Ali	usman@uni.edu.pk	3442345678	BSIT	4th	NULL	NULL	PR01	AI in Healthcare	Developer	Using ML for diagnostics
P003	Usman Ali	usman@uni.edu.pk	3007894512	BSIT	4th	NULL	NULL	PR01	AI in Healthcare	Developer	Using ML for diagnostics
P003	Usman Ali	usman@uni.edu.pk	3007894512	BSIT	4th	NULL	NULL	PR01	AI in Healthcare	Developer	Using ML for diagnostics
P004	Dr. Amna Tariq	amna@faculty.edu	3214567890	NULL	NULL	CS Dept	Machine Learning	PR03	Smart Agriculture	Supervisor	AI for crop yield prediction
P004	Dr. Amna Tariq	amna@faculty.edu	3214567890	NULL	NULL	CS Dept	Machine Learning	PR03	Smart Agriculture	Supervisor	AI for crop yield prediction
P005	Bilal Ahmed	bilal@student.com	3001112222	BSCS	1st	NULL	NULL	PR02	Secure IoT	Intern	Enhancing IoT security
P005	Bilal Ahmed	bilal@student.com	3001112222	BSCS	1st	NULL	NULL	PR02	Secure IoT	Intern	Enhancing IoT security
P006	Dr. Zeeshan Ali	zeeshan@uni.edu.pk	3228889999	NULL	NULL	IT Dept	Cloud Computing	PR04	Distributed Systems	Supervisor	Secure data across nodes
P007	Dr. Nadia Kamal	nadia@faculty.pk	3334445555	NULL	NULL	SE Dept	Software Architecture	PR05	Microservices Design	Lead	Optimizing Microservice Arch



Hour_Per_Week	Project_Start_Date	Project_End_Date	Funding_Body	Funding_Amount	Publication_ID
10	01/01/2023	01/06/2023	HEC	500000	PUB001
10	01/01/2023	01/06/2023	Ignite	500000	PUB001
10	01/01/2023	01/06/2023	HEC	500000	PUB001
10	01/01/2023	01/06/2023	Ignite	500000	PUB001
8	01/03/2023	01/09/2023	Ignite	300000	PUB002
8	01/03/2023	01/09/2023	Ignite	300000	PUB002
6	01/01/2023	01/06/2023	HEC	500000	PUB001
6	01/01/2023	01/06/2023	Ignite	500000	PUB001
6	01/01/2023	01/06/2023	HEC	500000	PUB001
6	01/01/2023	01/06/2023	Ignite	500000	PUB001
NULL	01/02/2023	01/08/2023	PARC	700000	PUB003
NULL	01/02/2023	01/08/2023	Ignite	700000	PUB003
5	01/03/2023	01/09/2023	Ignite	300000	PUB004
5	01/03/2023	01/09/2023	Ignite	300000	PUB004
NULL	02/01/2023	02/07/2023	HEC	600000	PUB005
NULL	01/04/2023	01/10/2023	Ignite	450000	PUB006

Publication_Title	Publication_Type	Publication_Year	Publisher	DOI	Author_Order
ML for Beginners	Journal	2023	IEEE	10.1109/xyz1	1
ML for Beginners	Journal	2023	Elsevier	10.1109/xyz1	2
ML for Beginners	Journal	2023	IEEE	10.1109/xyz1	1
ML for Beginners	Journal	2023	Elsevier	10.1109/xyz1	2
IoT Security Review	Conference	2023	ACM	10.1145/abc2	2
IoT Security Review	Conference	2023	ACM	10.1145/abc2	2
ML for Beginners	Journal	2023	IEEE	10.1109/xyz1	3
ML for Beginners	Journal	2023	Elsevier	10.1109/xyz1	3
ML for Beginners	Journal	2023	IEEE	10.1109/xyz1	3
ML for Beginners	Journal	2023	Elsevier	10.1109/xyz1	3
AgriTech Innovations	Book Chapter	2024	Springer	10.1007/agri3	1
AgriTech Innovations	Book Chapter	2024	Springer	10.1007/agri3	1
Intro to Embedded Sec.	Conference	2023	ACM	10.1145/emb4	2
Intro to Embedded Sec.	Conference	2023	ACM	10.1145/emb4	2
Secure Clouds	Journal	2023	IEEE	10.1109/cloud5	1
Designing Microservices	Book Chapter	2024	Springer	10.1007/msarch	1



Conference_ID	Conference_Name	Conference_Type	Conference_Date	Conference_Location	Conference_Status
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF003	AgriTech Meet	Regional	12/02/2024	Islamabad	Upcoming
CONF003	AgriTech Meet	Regional	12/02/2024	Islamabad	Upcoming
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF004	CloudConf	International	03/03/2023	Lahore	Active
CONF005	SE Symposium	National	20/04/2024	Islamabad	Upcoming

Conference_Role	Person_Type	Supervisor_ID	Project_Lead_ID
Presenter	Student	P004	P001
Organizer	Student	P004	P001
Presenter	Student	P004	P001
Organizer	Student	P004	P001
Presenter	Student	P004	P002
Presenter	Student	P004	P002
Attendee	Student	P004	P001
Attendee	Student	P004	P001
Attendee	Student	P004	P001
Attendee	Student	P004	P001
Keynote Speaker	Faculty	-	P004
Keynote Speaker	Faculty	-	P004
Attendee	Student	P004	P002
Organizer	Student	P004	P002
Session Chair	Faculty	-	P006
Speaker	Faculty	-	P007

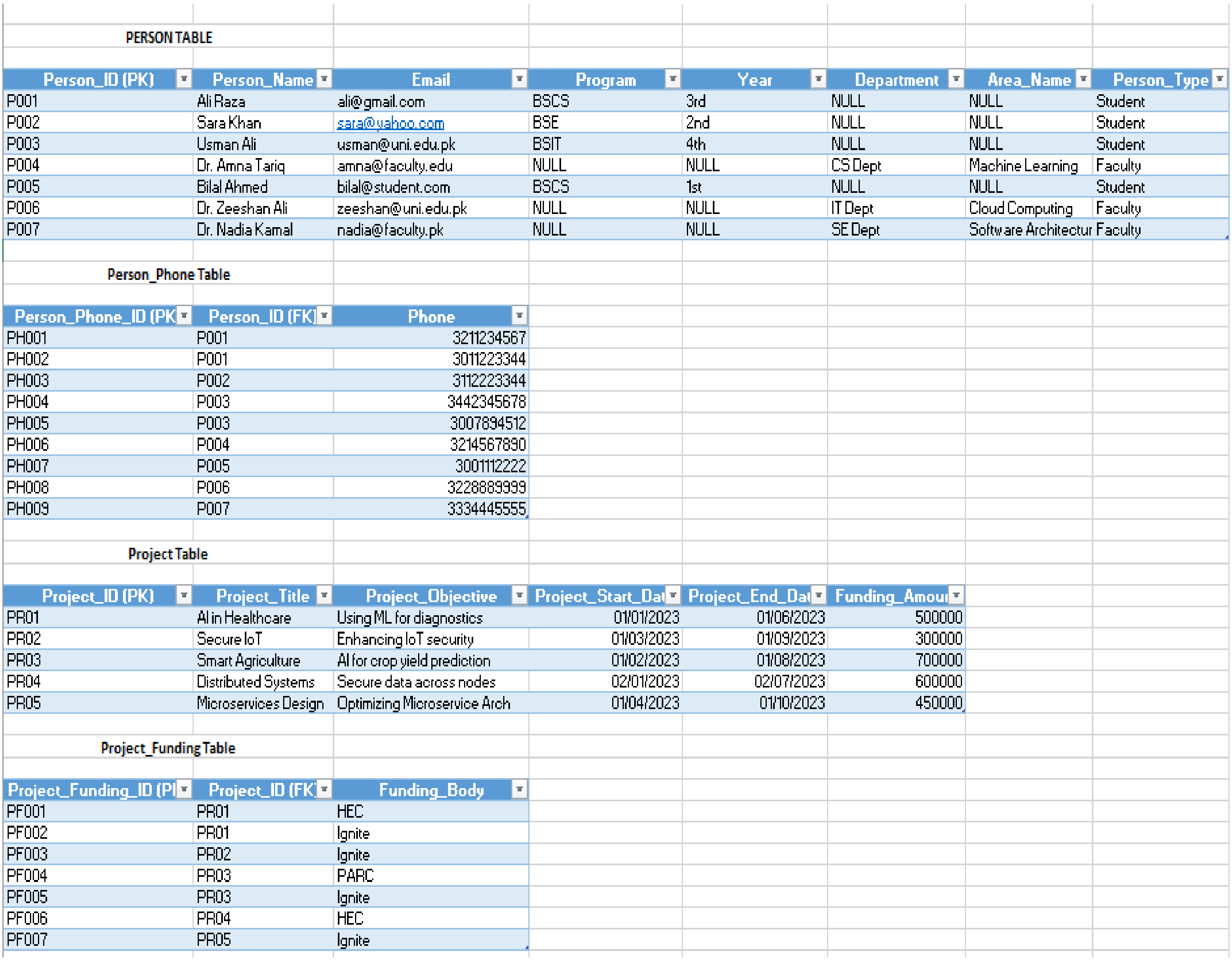
Normal Form (1NF) – Atomicity and Anomaly Resolution

In 1NF, all multi-valued fields (like Phone, Area_Name, Project_Role) were split into separate rows or separate related tables.This removed repeating groups and fixed the insertion, deletion, and update anomalies present in UNF.

For example, previously Ali Raza (P001) had two phone numbers in a single cell. In 1NF, both numbers were placed in separate rows in the Person_Phone table, allowing us to insert, delete, or update a phone number without affecting the rest of his project or publication data.

The data is now atomic and ready for 2NF.

Second Normal Form (2-NF) – Table Structure



PERSON TABLE

Person_ID (PK)	Person_Name	Email	Program	Year	Department	Area_Name	Person_Type
P001	Ali Raza	ali@gmail.com	BSCS	3rd	NULL	NULL	Student
P002	Sara Khan	sara@yahoo.com	BSE	2nd	NULL	NULL	Student
P003	Usman Ali	usman@uni.edu.pk	BSIT	4th	NULL	NULL	Student
P004	Dr. Amna Tariq	amna@faculty.edu	NULL	NULL	CS Dept	Machine Learning	Faculty
P005	Bilal Ahmed	bilal@student.com	BSCS	1st	NULL	NULL	Student
P006	Dr. Zeeshan Ali	zeeshan@uni.edu.pk	NULL	NULL	IT Dept	Cloud Computing	Faculty
P007	Dr. Nadia Kamal	nadia@faculty.pk	NULL	NULL	SE Dept	Software Architectur	Faculty

Person_Phone Table

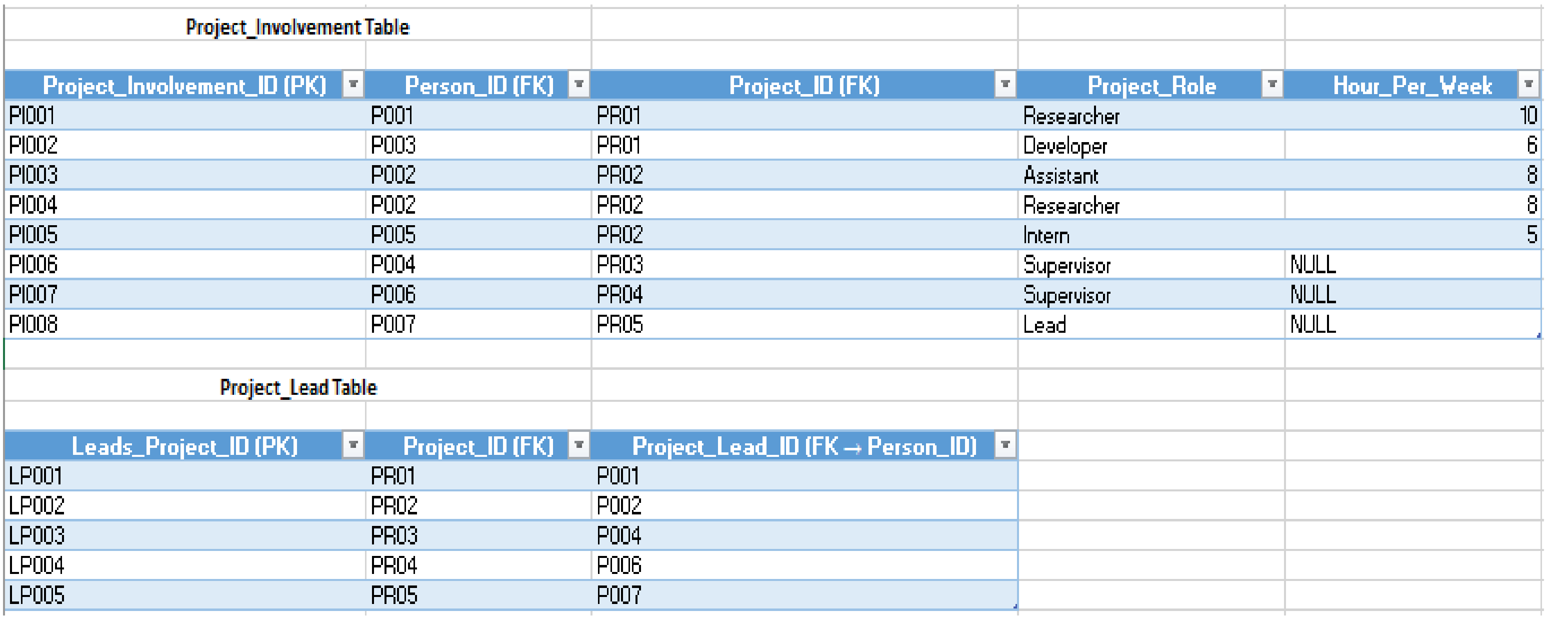
Person_Phone_ID (PK	Person_ID (FK)	Phone
PH001	P001	3211234567
PH002	P001	3011223344
PH003	P002	3112223344
PH004	P003	3442345678
PH005	P003	3007894512
PH006	P004	3214567890
PH007	P005	3001112222
PH008	P006	3228889999
PH009	P007	3334445555

Project Table

Project_ID (PK)	Project_Title	Project_Objective	Project_Start_Dat	Project_End_Dat	Funding_Amou
PR01	AI in Healthcare	Using ML for diagnostics	01/01/2023	01/06/2023	500000
PR02	Secure IoT	Enhancing IoT security	01/03/2023	01/09/2023	300000
PR03	Smart Agriculture	AI for crop yield prediction	01/02/2023	01/08/2023	700000
PR04	Distributed Systems	Secure data across nodes	02/01/2023	02/07/2023	600000
PR05	Microservices Design	Optimizing Microservice Arch	01/04/2023	01/10/2023	450000

Project_Funding Table

Project_Funding_ID (PI	Project_ID (FK	Funding_Body
PF001	PR01	HEC
PF002	PR01	Ignite
PF003	PR02	Ignite
PF004	PR03	PARC
PF005	PR03	Ignite
PF006	PR04	HEC
PF007	PR05	Ignite

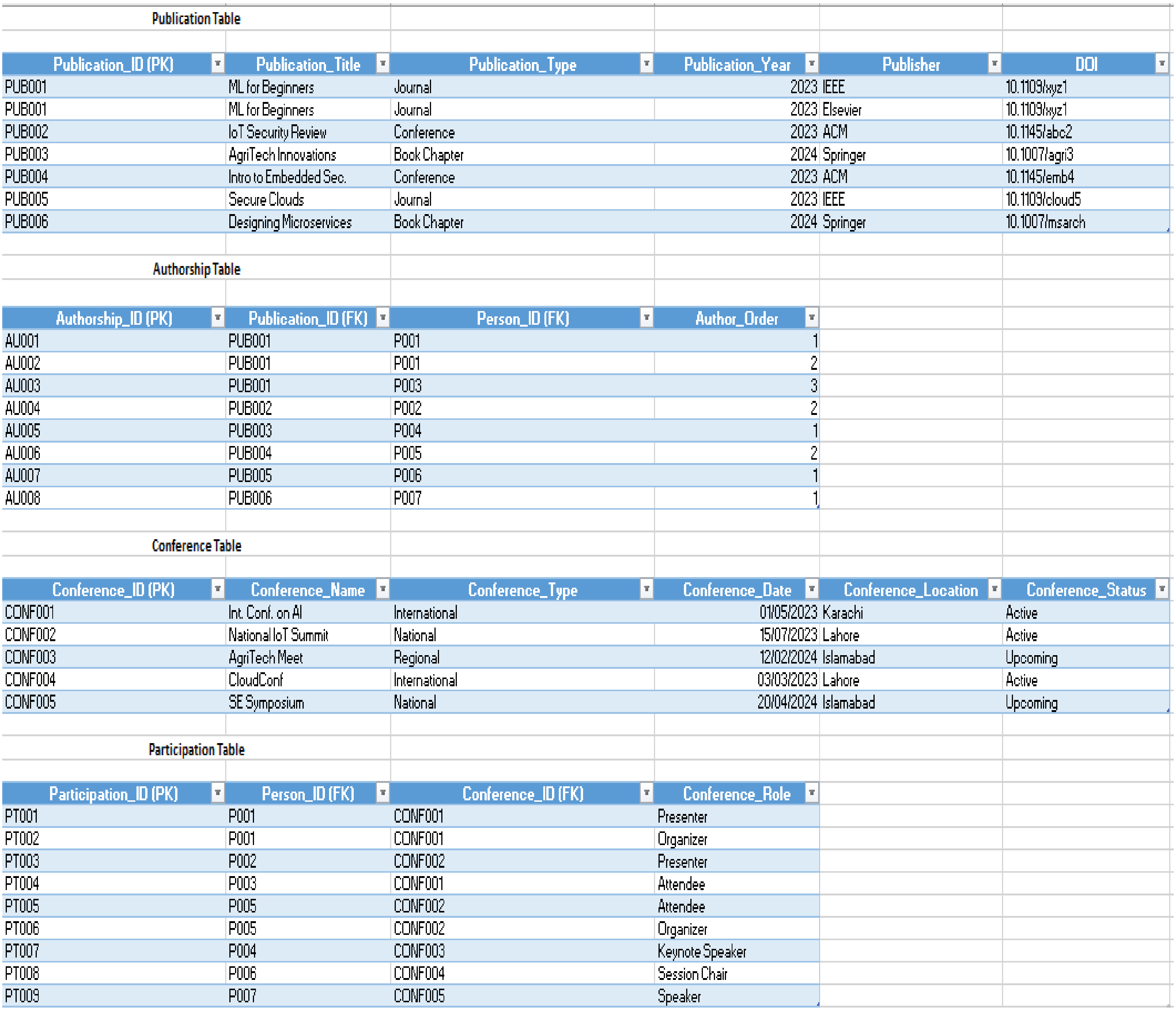


Project_Involvement Table

Project_Involvement_ID (PK)	Person_ID (FK)	Project_ID (FK)	Project_Role	Hour_Per_Week
PI001	P001	PR01	Researcher	10
PI002	P003	PR01	Developer	6
PI003	P002	PR02	Assistant	8
PI004	P002	PR02	Researcher	8
PI005	P005	PR02	Intern	5
PI006	P004	PR03	Supervisor	NULL
PI007	P006	PR04	Supervisor	NULL
PI008	P007	PR05	Lead	NULL

Project_Lead Table

Leads_Project_ID (PK)	Project_ID (FK)	Project_Lead_ID (FK → Person_ID)
LP001	PR01	P001
LP002	PR02	P002
LP003	PR03	P004
LP004	PR04	P006
LP005	PR05	P007



Publication Table

Publication_ID (PK)	Publication_Title	Publication_Type	Publication_Year	Publisher	DOI
PUB001	ML for Beginners	Journal	2023	IEEE	10.1109/xyz1
PUB001	ML for Beginners	Journal	2023	Elsevier	10.1109/xyz1
PUB002	IoT Security Review	Conference	2023	ACM	10.1145/abc2
PUB003	AgriTech Innovations	Book Chapter	2024	Springer	10.1007/agri3
PUB004	Intro to Embedded Sec.	Conference	2023	ACM	10.1145/emb4
PUB005	Secure Clouds	Journal	2023	IEEE	10.1109/cloud5
PUB006	Designing Microservices	Book Chapter	2024	Springer	10.1007/msarch

Authorship Table

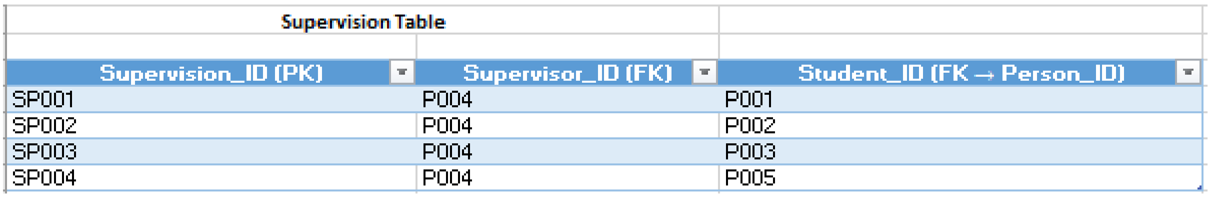
Authorship_ID (PK)	Publication_ID (FK)	Person_ID (FK)	Author_Order
AU001	PUB001	P001	1
AU002	PUB001	P001	2
AU003	PUB001	P003	3
AU004	PUB002	P002	2
AU005	PUB003	P004	1
AU006	PUB004	P005	2
AU007	PUB005	P006	1
AU008	PUB006	P007	1

Conference Table

Conference_ID (PK)	Conference_Name	Conference_Type	Conference_Date	Conference_Location	Conference_Status
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF003	AgriTech Meet	Regional	12/02/2024	Islamabad	Upcoming
CONF004	CloudConf	International	03/03/2023	Lahore	Active
CONF005	SE Symposium	National	20/04/2024	Islamabad	Upcoming

Participation Table

Participation_ID (PK)	Person_ID (FK)	Conference_ID (FK)	Conference_Role
PT001	P001	CONF001	Presenter
PT002	P001	CONF001	Organizer
PT003	P002	CONF002	Presenter
PT004	P003	CONF001	Attendee
PT005	P005	CONF002	Attendee
PT006	P005	CONF002	Organizer
PT007	P004	CONF003	Keynote Speaker
PT008	P006	CONF004	Session Chair
PT009	P007	CONF005	Speaker



Supervision Table

Supervision_ID (PK)	Supervisor_ID (FK)	Student_ID (FK → Person_ID)
SP001	P004	P001
SP002	P004	P002
SP003	P004	P003
SP004	P004	P005

Second Normal Form (2NF) – Removal of Partial Dependencies

In 2NF, partial dependencies were removed by splitting data into separate tables.

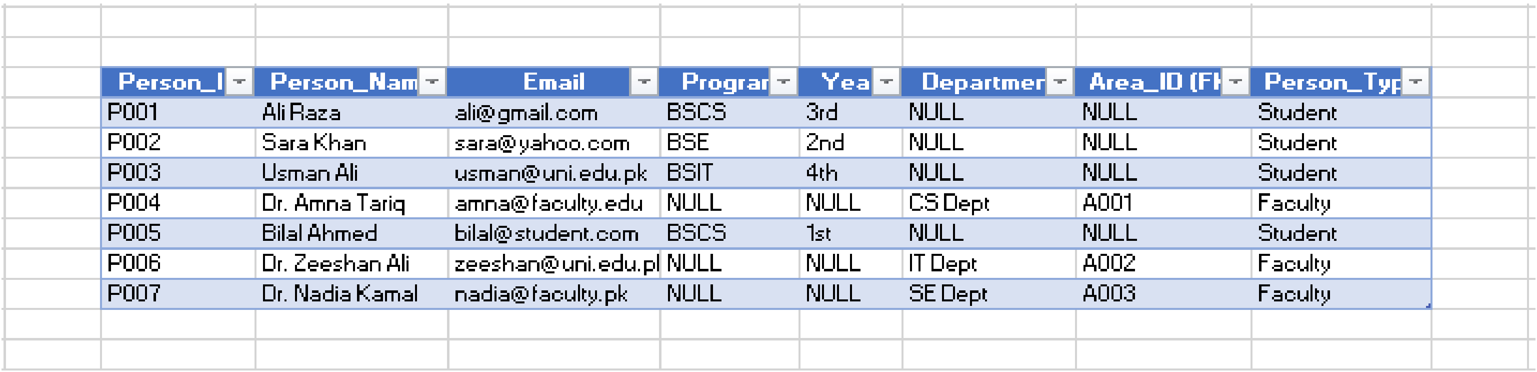
For example, in the UNF and 1NF tables, the Project_Role like Researcher or Intern depended only on the Project_ID and not the full composite key (Person_ID + Project_ID). This was moved to a new table: Person_Project_Role(Person_ID, Project_ID, Role), which makes the structure cleaner and avoids repeating role data with every person or project.Similarly, Phone numbers and Area_Name depended only on Person_ID, so they were moved into separate tables:

Person_Phone(Person_ID, Phone)

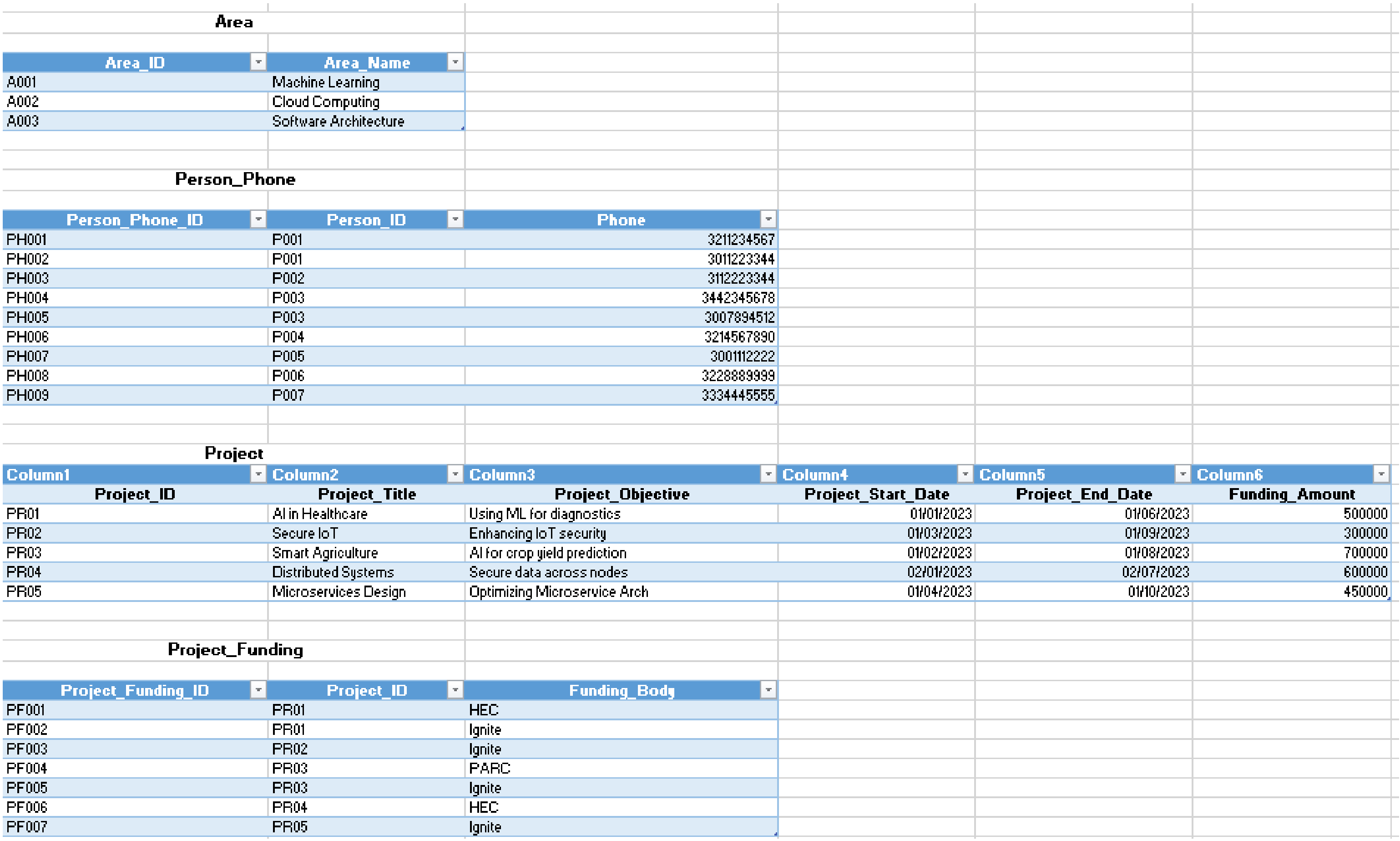
Person_Area(Person_ID, Area_Name)

Now, each non-key attribute fully depends on the entire primary key, not part of it. This resolved redundancy and prepared the design for 3NF.

Third Normal Form (3-NF) – Table Structure



Person_I	Person_Nam	Email	Prograr	Yea	Departmer	Area_ID (FI	Person_Typ
P001	Ali Raza	ali@gmail.com	BSCS	3rd	NULL	NULL	Student
P002	Sara Khan	sara@yahoo.com	BSE	2nd	NULL	NULL	Student
P003	Usman Ali	usman@uni.edu.pk	BSIT	4th	NULL	NULL	Student
P004	Dr. Amna Tariq	amna@faculty.edu	NULL	NULL	CS Dept	A001	Faculty
P005	Bilal Ahmed	bilal@student.com	BSCS	1st	NULL	NULL	Student
P006	Dr. Zeeshan Ali	zeeshan@uni.edu.pl	NULL	NULL	IT Dept	A002	Faculty
P007	Dr. Nadia Kamal	nadia@faculty.pk	NULL	NULL	SE Dept	A003	Faculty



Area

Area_ID	Area_Name
A001	Machine Learning
A002	Cloud Computing
A003	Software Architecture

Person_Phone

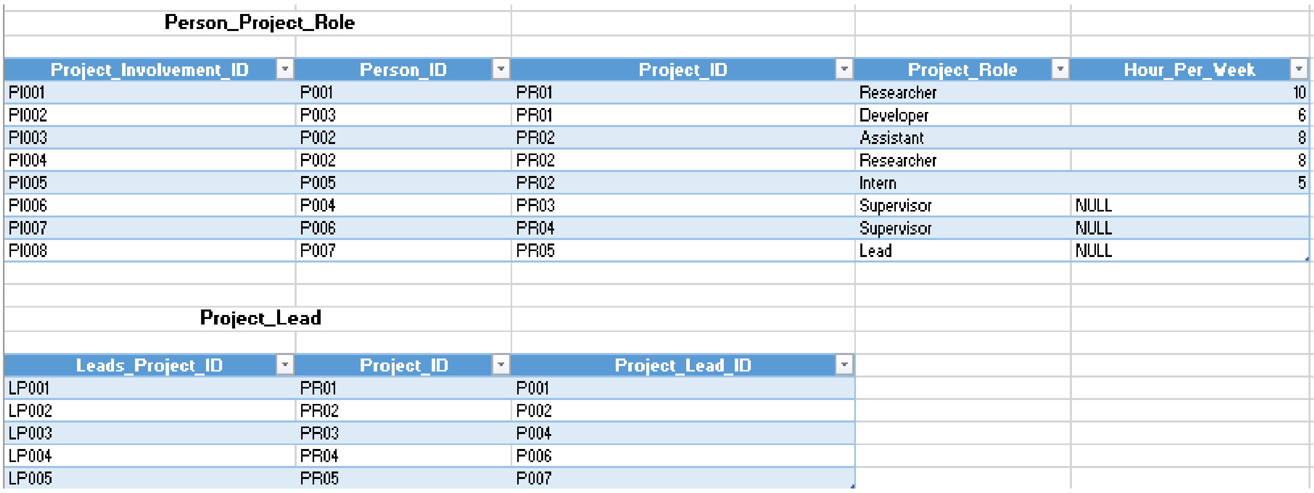
Person_Phone_ID	Person_ID	Phone
PH001	P001	3211234567
PH002	P001	3011223344
PH003	P002	3112223344
PH004	P003	3442345678
PH005	P003	3007894512
PH006	P004	3214567890
PH007	P005	3001112222
PH008	P006	3228889999
PH009	P007	3334445555

Project

Column1	Column2	Column3	Column4	Column5	Column6
Project_ID	Project_Title	Project_Objective	Project_Start_Date	Project_End_Date	Funding_Amount
PR01	AI in Healthcare	Using ML for diagnostics	01/01/2023	01/06/2023	500000
PR02	Secure IoT	Enhancing IoT security	01/03/2023	01/09/2023	300000
PR03	Smart Agriculture	AI for crop yield prediction	01/02/2023	01/08/2023	700000
PR04	Distributed Systems	Secure data across nodes	02/01/2023	02/07/2023	600000
PR05	Microservices Design	Optimizing Microservice Arch	01/04/2023	01/10/2023	450000

Project_Funding

Project_Funding_ID	Project_ID	Funding_Body
PF001	PR01	HEC
PF002	PR01	Ignite
PF003	PR02	Ignite
PF004	PR03	PARC
PF005	PR03	Ignite
PF006	PR04	HEC
PF007	PR05	Ignite



Person_Project_Role

Project_Involvement_ID	Person_ID	Project_ID	Project_Role	Hour_Per_Week
PI001	P001	PR01	Researcher	10
PI002	P003	PR01	Developer	6
PI003	P002	PR02	Assistant	8
PI004	P002	PR02	Researcher	8
PI005	P005	PR02	Intern	5
PI006	P004	PR03	Supervisor	NULL
PI007	P006	PR04	Supervisor	NULL
PI008	P007	PR05	Lead	NULL

Project_Lead

Leads_Project_ID	Project_ID	Project_Lead_ID
LP001	PR01	P001
LP002	PR02	P002
LP003	PR03	P004
LP004	PR04	P006
LP005	PR05	P007

Publisher

Publisher_ID	Publisher_Name
PUB001	IEEE
PUB002	Elsevier
PUB003	ACM
PUB004	Springer

Publication

Publication_ID	Publication_Title	Publication_Type	Publication_Year	Publisher_ID	DOI
PUB001	ML for Beginners	Journal	2023	PUB001	10.1109/xyz1
PUB001	ML for Beginners	Journal	2023	PUB002	10.1109/xyz1
PUB002	IoT Security Review	Conference	2023	PUB003	10.1145/abc2
PUB003	AgriTech Innovations	Book Chapter	2024	PUB004	10.1007/agri3
PUB004	Intro to Embedded Sec.	Conference	2023	PUB003	10.1145/emb4
PUB005	Secure Clouds	Journal	2023	PUB001	10.1109/cloud5
PUB006	Designing Microservices	Book Chapter	2024	PUB004	10.1007/msarch

Publication_Authorship

Authorship_ID	Publication_ID	Person_ID	Author_Order
AU001	PUB001	P001	1
AU002	PUB001	P001	2
AU003	PUB001	P003	3
AU004	PUB002	P002	2
AU005	PUB003	P004	1
AU006	PUB004	P005	2
AU007	PUB005	P006	1
AU008	PUB006	P007	1

Conference

Conference_ID	Conference_Name	Conference_Type	Conference_Date	Conference_Location	Conference_Status
CONF001	Int. Conf. on AI	International	01/05/2023	Karachi	Active
CONF002	National IoT Summit	National	15/07/2023	Lahore	Active
CONF003	AgriTech Meet	Regional	12/02/2024	Islamabad	Upcoming
CONF004	CloudConf	International	03/03/2023	Lahore	Active
CONF005	SE Symposium	National	20/04/2024	Islamabad	Upcoming

Role

Role_ID	Role_Name
R001	Presenter
R002	Organizer
R003	Attendee
R004	Keynote Speaker
R005	Session Chair
R006	Speaker

Conference_Participation

Participation_ID	Person_ID	Conference_ID	Role_ID
PT001	P001	CONF001	R001
PT002	P001	CONF001	R002
PT003	P002	CONF002	R001
PT004	P003	CONF001	R003
PT005	P005	CONF002	R003
PT006	P005	CONF002	R002
PT007	P004	CONF003	R004
PT008	P006	CONF004	R005
PT009	P007	CONF005	R006

Supervision

Supervision_ID	Supervisor_ID	Student_ID
SP001	P004	P001
SP002	P004	P002
SP003	P004	P003
SP004	P004	P005

Third Normal Form (3NF) – Removal of Transitive Dependencies

In 3NF, all transitive dependencies were removed.

For example, in the Person (Student) table, the Department information was separated into its own table because it depended on another attribute (e.g., Program) instead of depending directly on the primary key Person_ID.

Similarly, attributes such as roles, publishers, and research areas — which originally depended indirectly through other non-key attributes — were also placed in separate tables.

Now, every non-key attribute in every table depends **only** on the primary key of that table and **not on any other non-key attribute**.

structure is now fully normalized and ready for implementation.

Physical Schema – SQL Table Definitions And Creation

Table Creation In SQL

This section presents the 14 finalized tables for the Research and Conference Management System, structured in 3rd Normal Form (3NF). Each table includes appropriate constraints like Primary Keys, Foreign Keys, NOT NULL, and UNIQUE to ensure data integrity. The schema is based on a normalized ER model and was successfully implemented in Oracle SQL Developer to support all defined business rules.

```
--1. Area
CREATE TABLE Area (
Area_ID VARCHAR2(10) PRIMARY KEY,
Area_Name VARCHAR2(100) NOT NULL
);
```

```
--2. Person
CREATE TABLE Person (
Person_ID VARCHAR2(10) PRIMARY KEY,
Person_Name VARCHAR2(100) NOT NULL,
Email VARCHAR2(100) UNIQUE,
Program VARCHAR2(50),
Year VARCHAR2(20),
Department VARCHAR2(100),
Area_ID VARCHAR2(10),
Person_Type VARCHAR2(20) NOT NULL CHECK (Person_Type IN ('Student','Faculty')),
CONSTRAINT fk_person_area FOREIGN KEY (Area_ID) REFERENCES Area(Area_ID)
);
```

```
--3. Person_Phone

CREATE TABLE Person_Phone (

Person_Phone_ID VARCHAR2(10) PRIMARY KEY,

Person_ID VARCHAR2(10) NOT NULL,

Phone VARCHAR2(20) NOT NULL,

CONSTRAINT fk_phone_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID)

);
```

```
--4. Project

CREATE TABLE Project (

Project_ID VARCHAR2(10) PRIMARY KEY,

Project_Title VARCHAR2(200) NOT NULL,

Project_Objective VARCHAR2(500),

Project_Start_Date DATE NOT NULL,

Project_End_Date DATE,

Funding_Amount NUMBER(12,2)

);
```

```
--5. Project_Funding

CREATE TABLE Project_Funding (

Project_Funding_ID VARCHAR2(10) PRIMARY KEY,

Project_ID VARCHAR2(10) NOT NULL,

Funding_Body VARCHAR2(100) NOT NULL,

CONSTRAINT fk_funding_project FOREIGN KEY (Project_ID) REFERENCES Project(Project_ID)

);
```

--6. Person_Project_Role

```
CREATE TABLE Person_Project_Role (
Project_Involvement_ID VARCHAR2(10) PRIMARY KEY,
Person_ID VARCHAR2(10) NOT NULL,
Project_ID VARCHAR2(10) NOT NULL,
Project_Role VARCHAR2(100),
Hour_Per_Week NUMBER,
CONSTRAINT fk_role_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),
CONSTRAINT fk_role_project FOREIGN KEY (Project_ID) REFERENCES Project(Project_ID)
);
```

--7. Project_Lead

```
CREATE TABLE Project_Lead (
Leads_Project_ID VARCHAR2(10) PRIMARY KEY,
Project_ID VARCHAR2(10) NOT NULL UNIQUE,
Project_Lead_ID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_lead_project FOREIGN KEY (Project_ID) REFERENCES Project(Project_ID),
CONSTRAINT fk_lead_person FOREIGN KEY (Project_Lead_ID) REFERENCES Person(Person_ID)
);
```

--8. Publisher

```
CREATE TABLE Publisher (
Publisher_ID VARCHAR2(10) PRIMARY KEY,
Publisher_Name VARCHAR2(100) NOT NULL
);
```

--9. Publication

```
CREATE TABLE Publication (
Publication_ID VARCHAR2(10) PRIMARY KEY,
Publication_Title VARCHAR2(200) NOT NULL,
Publication_Type VARCHAR2(50),
Publication_Year NUMBER(4),
Publisher_ID VARCHAR2(10) NOT NULL,
DOI VARCHAR2(100),
CONSTRAINT fk_pub_publisher FOREIGN KEY (Publisher_ID) REFERENCES Publisher(Publisher_ID)
);
```

--10. Publication_Authorship

```
CREATE TABLE Publication_Authorship (
Authorship_ID VARCHAR2(10) PRIMARY KEY,
Publication_ID VARCHAR2(10) NOT NULL,
Person_ID VARCHAR2(10) NOT NULL,
Author_Order NUMBER NOT NULL,
CONSTRAINT fk_authorship_pub FOREIGN KEY (Publication_ID) REFERENCES Publication(Publication_ID),
CONSTRAINT fk_authorship_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID)
);
```

--11. Conference

```
CREATE TABLE Conference (

Conference_ID VARCHAR2(10) PRIMARY KEY,

Conference_Name VARCHAR2(200) NOT NULL,

Conference_Type VARCHAR2(50),

Conference_Date DATE NOT NULL,

Conference_Location VARCHAR2(100),

Conference_Status VARCHAR2(20) CHECK (Conference_Status IN ('Active','Upcoming','Completed'))

);
```

--12. Role

```
CREATE TABLE Role (

Role_ID VARCHAR2(10) PRIMARY KEY,

Role_Name VARCHAR2(100) NOT NULL

);
```

--13. Conference_Participation

```
CREATE TABLE Conference_Participation (

Participation_ID VARCHAR2(10) PRIMARY KEY,

Person_ID VARCHAR2(10) NOT NULL,

Conference_ID VARCHAR2(10) NOT NULL,

Role_ID VARCHAR2(10) NOT NULL,

CONSTRAINT fk_part_person FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),

CONSTRAINT fk_part_conf FOREIGN KEY (Conference_ID) REFERENCES Conference(Conference_ID),

CONSTRAINT fk_part_role FOREIGN KEY (Role_ID) REFERENCES Role(Role_ID)

);
```

```
--14. Supervision
CREATE TABLE Supervision (
Supervision_ID VARCHAR2(10) PRIMARY KEY,
Supervisor_ID VARCHAR2(10) NOT NULL,
Student_ID VARCHAR2(10) NOT NULL,
CONSTRAINT fk_sup_supervisor FOREIGN KEY (Supervisor_ID) REFERENCES Person(Person_ID),
CONSTRAINT fk_sup_student FOREIGN KEY (Student_ID) REFERENCES Person(Person_ID)
);
```

Data Insertion In SQL

Sample data was inserted into all 14 tables to validate the design and ensure constraint enforcement. The INSERT statements cover key relationships—one-to-many and many-to-many—and reflect real project scenarios like multiple funding bodies, roles per person, and publication authorships. All insertions were successful without any constraint violations.

--1. Area

```
INSERT INTO Area (Area_ID, Area_Name) VALUES ('A001', 'Machine Learning');

INSERT INTO Area (Area_ID, Area_Name) VALUES ('A002', 'Cloud Computing');

INSERT INTO Area (Area_ID, Area_Name) VALUES ('A003', 'Software Architecture');
```

-- 2. Person

```
INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P001','Ali Raza','ali@gmail.com','BSCS','3rd',NULL,NULL,'Student');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P002','Sara Khan','sara@yahoo.com','BSE','2nd',NULL,NULL,'Student');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P003','Usman Ali','usman@uni.edu.pk','BSIT','4th',NULL,NULL,'Student');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P004','Dr. Amna Tariq','amna@faculty.edu',NULL,NULL,'CS Dept','A001','Faculty');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P005','Bilal Ahmed','bilal@student.com','BSCS','1st',NULL,NULL,'Student');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)

VALUES ('P006','Dr. Zeeshan Ali','zeeshan@uni.edu.pk',NULL,NULL,'IT Dept','A002','Faculty');

INSERT INTO Person (Person_ID, Person_Name, Email, Program, Year, Department, Area_ID, Person_Type)
```

```
VALUES ('P007','Dr. Nadia Kamal','nadia@faculty.pk',NULL,NULL,'SE Dept','A003','Faculty');
```

-- 3. Person_Phone

```
INSERT INTO Person_Phone VALUES ('PH001','P001','3211234567');

INSERT INTO Person_Phone VALUES ('PH002','P001','3011223344');

INSERT INTO Person_Phone VALUES ('PH003','P002','3112223344');

INSERT INTO Person_Phone VALUES ('PH004','P003','3442345678');

INSERT INTO Person_Phone VALUES ('PH005','P003','3007894512');

INSERT INTO Person_Phone VALUES ('PH006','P004','3214567890');

INSERT INTO Person_Phone VALUES ('PH007','P005','3001112222');

INSERT INTO Person_Phone VALUES ('PH008','P006','3228889999');

INSERT INTO Person_Phone VALUES ('PH009','P007','3334445555');
```

-- 4. Project

```
INSERT INTO Project VALUES ('PR01','AI in Healthcare','Using ML for diagnostics',TO_DATE('01/01/2023','DD/MM/YYYY'),TO_DATE('01/06/2023','DD/MM/YYYY'),500000);

INSERT INTO Project VALUES ('PR02','Secure IoT','Enhancing IoT security',TO_DATE('01/03/2023','DD/MM/YYYY'),TO_DATE('01/09/2023','DD/MM/YYYY'),300000);

INSERT INTO Project VALUES ('PR03','Smart Agriculture','AI for crop yield prediction',TO_DATE('01/02/2023','DD/MM/YYYY'),TO_DATE('01/08/2023','DD/MM/YYYY'),700000);

INSERT INTO Project VALUES ('PR04','Distributed Systems','Secure data across nodes',TO_DATE('02/01/2023','DD/MM/YYYY'),TO_DATE('02/07/2023','DD/MM/YYYY'),600000);

INSERT INTO Project VALUES ('PR05','Microservices Design','Optimizing Microservice Arch',TO_DATE('01/04/2023','DD/MM/YYYY'),TO_DATE('01/10/2023','DD/MM/YYYY'),450000);
```

```sql
-- 5. Project_Funding
INSERT INTO Project_Funding VALUES ('PF001','PR01','HEC');
INSERT INTO Project_Funding VALUES ('PF002','PR01','Ignite');
INSERT INTO Project_Funding VALUES ('PF003','PR02','Ignite');
INSERT INTO Project_Funding VALUES ('PF004','PR03','PARC');
INSERT INTO Project_Funding VALUES ('PF005','PR03','Ignite');
INSERT INTO Project_Funding VALUES ('PF006','PR04','HEC');
INSERT INTO Project_Funding VALUES ('PF007','PR05','Ignite');

-- 6. Person_Project_Role
INSERT INTO Person_Project_Role VALUES ('PI001','P001','PR01','Researcher',10);
INSERT INTO Person_Project_Role VALUES ('PI002','P003','PR01','Developer',6);
INSERT INTO Person_Project_Role VALUES ('PI003','P002','PR02','Assistant',8);
INSERT INTO Person_Project_Role VALUES ('PI004','P002','PR02','Researcher',8);
INSERT INTO Person_Project_Role VALUES ('PI005','P005','PR02','Intern',5);
INSERT INTO Person_Project_Role VALUES ('PI006','P004','PR03','Supervisor',NULL);
INSERT INTO Person_Project_Role VALUES ('PI007','P006','PR04','Supervisor',NULL);
INSERT INTO Person_Project_Role VALUES ('PI008','P007','PR05','Lead',NULL);

-- 7. Project_Lead
INSERT INTO Project_Lead VALUES ('LP001','PR01','P001');
INSERT INTO Project_Lead VALUES ('LP002','PR02','P002');
INSERT INTO Project_Lead VALUES ('LP003','PR03','P004');
INSERT INTO Project_Lead VALUES ('LP004','PR04','P006');
INSERT INTO Project_Lead VALUES ('LP005','PR05','P007');
```

-- 8. Publisher

```
INSERT INTO Publisher VALUES ('PUB001','IEEE');

INSERT INTO Publisher VALUES ('PUB002','Elsevier');

INSERT INTO Publisher VALUES ('PUB003','ACM');

INSERT INTO Publisher VALUES ('PUB004','Springer');
```

-- 9. Publication

```
-- Note: Publication_ID must be unique, duplicates not allowed.

-- The second row (PUB001, PUB002) with same PUB001 will conflict, so adjust ID if needed:

INSERT INTO Publication VALUES ('PUB001','ML for Beginners','Journal',2023,'PUB001','10.1109/xyz1');

INSERT INTO Publication VALUES ('PUB007','ML for Beginners','Journal',2023,'PUB002','10.1109/xyz1');

INSERT INTO Publication VALUES ('PUB002','IoT Security Review','Conference',2023,'PUB003','10.1145/abc2');

INSERT INTO Publication VALUES ('PUB003','AgriTech Innovations','Book Chapter',2024,'PUB004','10.1007/agri3');

INSERT INTO Publication VALUES ('PUB004','Intro to Embedded Sec.','Conference',2023,'PUB003','10.1145/emb4');

INSERT INTO Publication VALUES ('PUB005','Secure Clouds','Journal',2023,'PUB001','10.1109/cloud5');

INSERT INTO Publication VALUES ('PUB006','Designing Microservices','Book Chapter',2024,'PUB004','10.1007/msarch');
```

```
-- 10. Publication_Authorship
INSERT INTO Publication_Authorship VALUES ('AU001','PUB001','P001',1);
INSERT INTO Publication_Authorship VALUES ('AU002','PUB001','P001',2);
INSERT INTO Publication_Authorship VALUES ('AU003','PUB001','P003',3);
INSERT INTO Publication_Authorship VALUES ('AU004','PUB002','P002',2);
INSERT INTO Publication_Authorship VALUES ('AU005','PUB003','P004',1);
INSERT INTO Publication_Authorship VALUES ('AU006','PUB004','P005',2);
INSERT INTO Publication_Authorship VALUES ('AU007','PUB005','P006',1);
INSERT INTO Publication_Authorship VALUES ('AU008','PUB006','P007',1);

-- 11. Conference
INSERT INTO Conference VALUES ('CONF001','Int. Conf. on AI','International',TO_DATE('01/05/2023','DD/MM/YYYY'),'Karachi','Active');
INSERT INTO Conference VALUES ('CONF002','National IoT Summit','National',TO_DATE('15/07/2023','DD/MM/YYYY'),'Lahore','Active');
INSERT INTO Conference VALUES ('CONF003','AgriTech Meet','Regional',TO_DATE('12/02/2024','DD/MM/YYYY'),'Islamabad','Upcoming');
INSERT INTO Conference VALUES ('CONF004','CloudConf','International',TO_DATE('03/03/2023','DD/MM/YYYY'),'Lahore','Active');
INSERT INTO Conference VALUES ('CONF005','SE Symposium','National',TO_DATE('20/04/2024','DD/MM/YYYY'),'Islamabad','Upcoming');

-- 12. Role
INSERT INTO Role VALUES ('R001','Presenter');
INSERT INTO Role VALUES ('R002','Organizer');
INSERT INTO Role VALUES ('R003','Attendee');
INSERT INTO Role VALUES ('R004','Keynote Speaker');
INSERT INTO Role VALUES ('R005','Session Chair');
INSERT INTO Role VALUES ('R006','Speaker');
```

-- 13. Conference_Participation

```
INSERT INTO Conference_Participation VALUES ('PT001','P001','CONF001','R001');
INSERT INTO Conference_Participation VALUES ('PT002','P001','CONF001','R002');
INSERT INTO Conference_Participation VALUES ('PT003','P002','CONF002','R001');
INSERT INTO Conference_Participation VALUES ('PT004','P003','CONF001','R003');
INSERT INTO Conference_Participation VALUES ('PT005','P005','CONF002','R003');
INSERT INTO Conference_Participation VALUES ('PT006','P005','CONF002','R002');
INSERT INTO Conference_Participation VALUES ('PT007','P004','CONF003','R004');
INSERT INTO Conference_Participation VALUES ('PT008','P006','CONF004','R005');
INSERT INTO Conference_Participation VALUES ('PT009','P007','CONF005','R006');
```

-- 14. Supervision

```
INSERT INTO Supervision VALUES ('SP001','P004','P001');
INSERT INTO Supervision VALUES ('SP002','P004','P002');
INSERT INTO Supervision VALUES ('SP003','P004','P003');
INSERT INTO Supervision VALUES ('SP004','P004','P005');
```

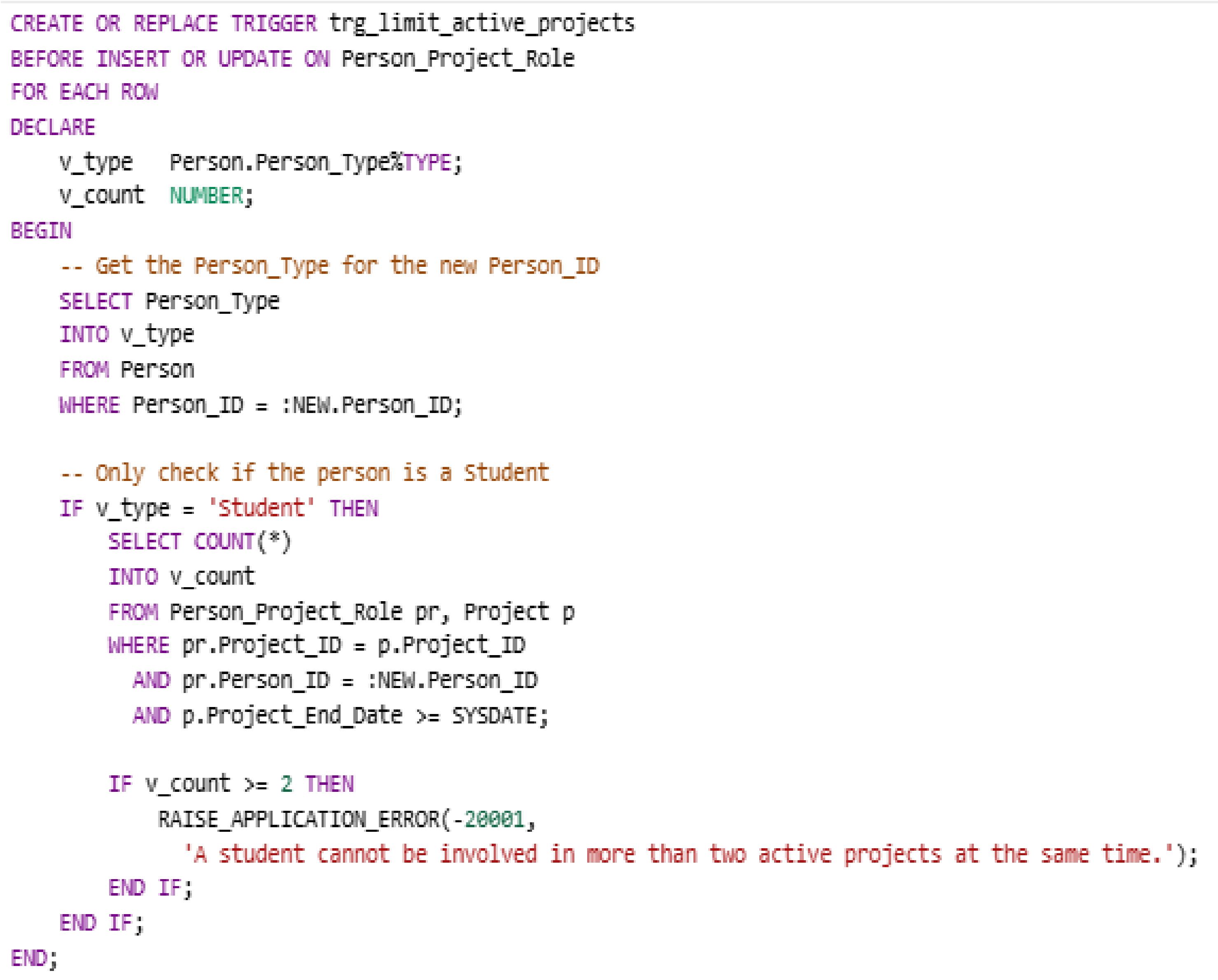
Triggers for Complex Constraints

Some business rules go beyond standard constraints, so Oracle PL/SQL triggers were implemented to enforce them. These triggers maintain data integrity by validating conditions during insert or update operations:

Trigger 1 – Limits each student to two active projects.
Trigger 2 – Requires at least one faculty author per publication.
Trigger 3 – Activates a conference only if it has participants.
Trigger 4 – Ensures a person is either a student or faculty, not both.

All triggers were successfully tested, and any rule violation raises a custom error message.

Trigger 1: Limit student to max 2 active projects

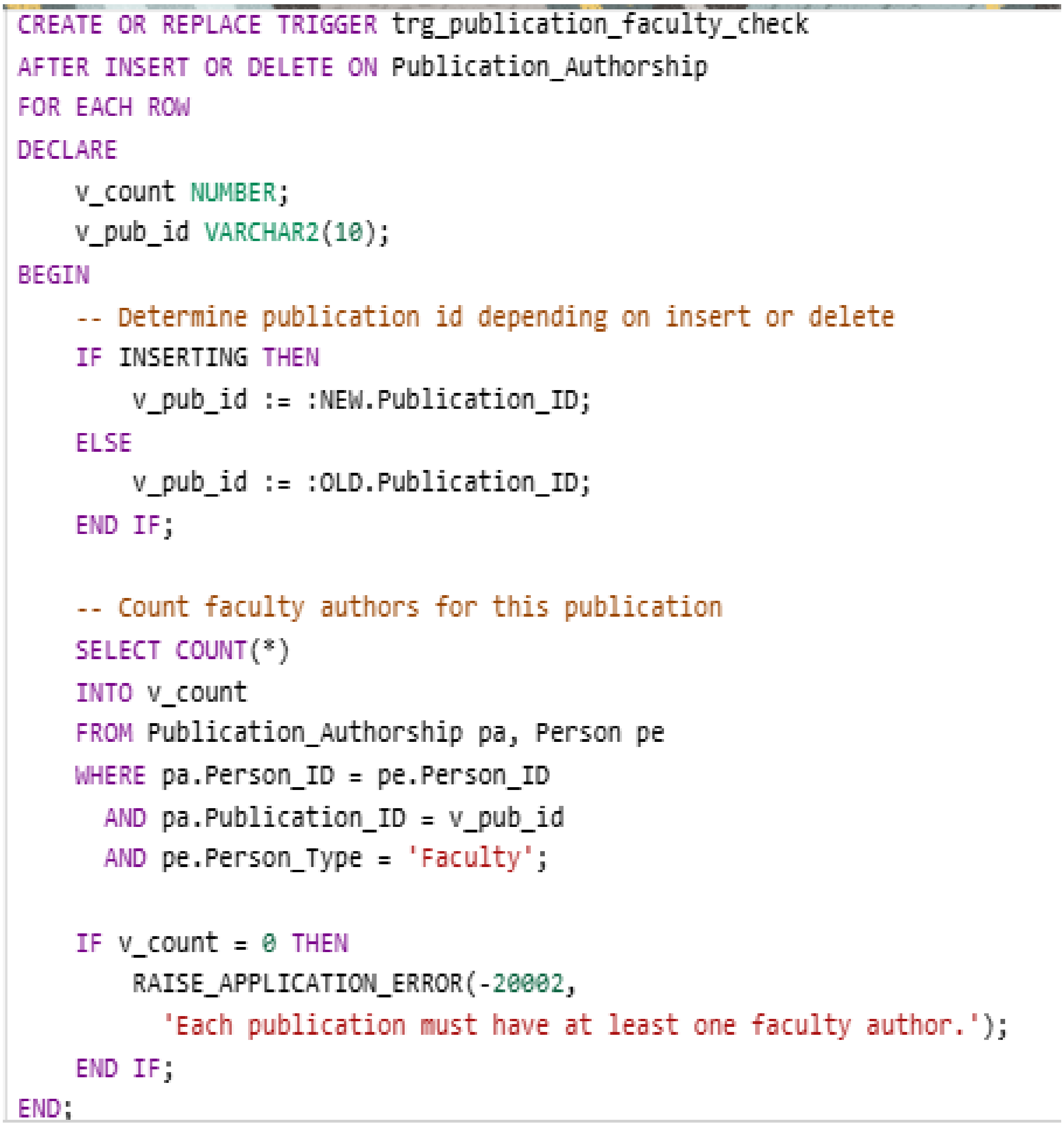


```sql
CREATE OR REPLACE TRIGGER trg_limit_active_projects
BEFORE INSERT OR UPDATE ON Person_Project_Role
FOR EACH ROW
DECLARE
    v_type   Person.Person_Type%TYPE;
    v_count  NUMBER;
BEGIN
    -- Get the Person_Type for the new Person_ID
    SELECT Person_Type
    INTO v_type
    FROM Person
    WHERE Person_ID = :NEW.Person_ID;

    -- Only check if the person is a Student
    IF v_type = 'Student' THEN
        SELECT COUNT(*)
        INTO v_count
        FROM Person_Project_Role pr, Project p
        WHERE pr.Project_ID = p.Project_ID
          AND pr.Person_ID = :NEW.Person_ID
          AND p.Project_End_Date >= SYSDATE;

        IF v_count >= 2 THEN
            RAISE_APPLICATION_ERROR(-20001,
              'A student cannot be involved in more than two active projects at the same time.');
        END IF;
    END IF;
END;
```

Trigger 2: Every publication must have at least one faculty author

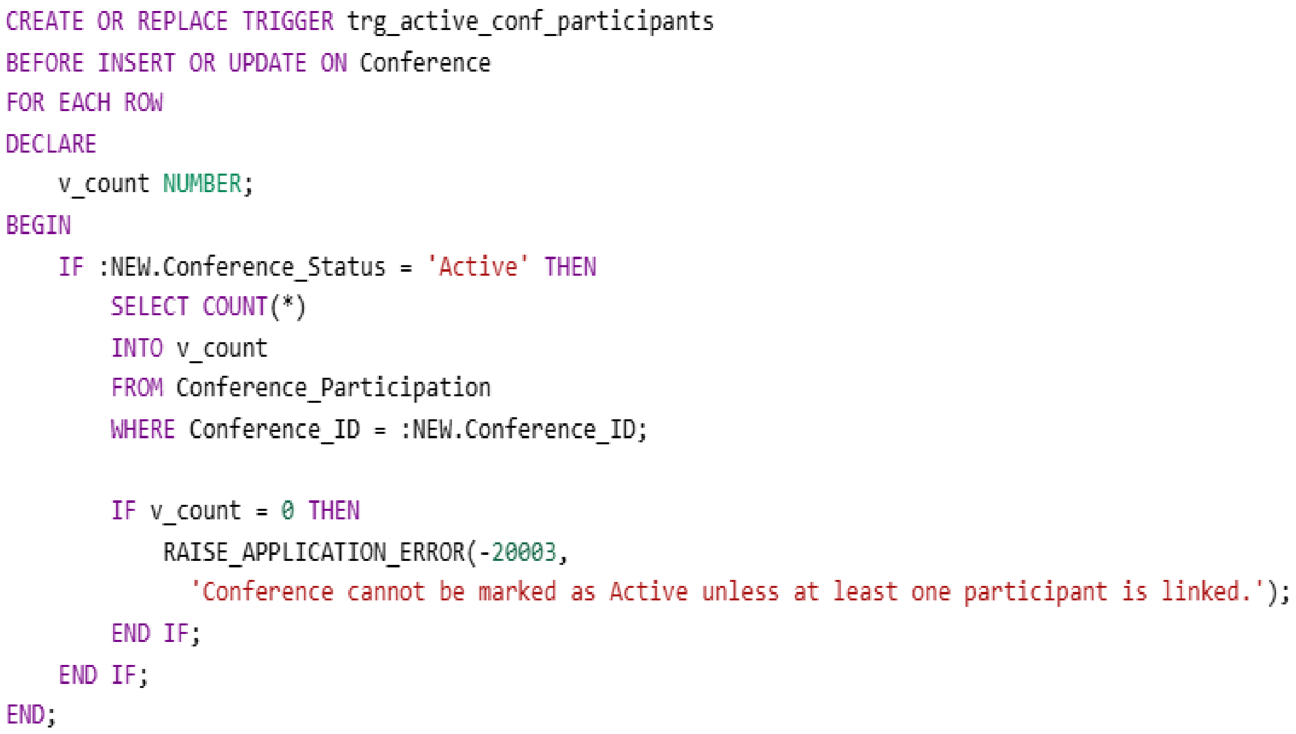


```
CREATE OR REPLACE TRIGGER trg_publication_faculty_check
AFTER INSERT OR DELETE ON Publication_Authorship
FOR EACH ROW
DECLARE
    v_count NUMBER;
    v_pub_id VARCHAR2(10);
BEGIN
    -- Determine publication id depending on insert or delete
    IF INSERTING THEN
        v_pub_id := :NEW.Publication_ID;
    ELSE
        v_pub_id := :OLD.Publication_ID;
    END IF;

    -- Count faculty authors for this publication
    SELECT COUNT(*)
    INTO v_count
    FROM Publication_Authorship pa, Person pe
    WHERE pa.Person_ID = pe.Person_ID
      AND pa.Publication_ID = v_pub_id
      AND pe.Person_Type = 'Faculty';

    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20002,
          'Each publication must have at least one faculty author.');
    END IF;
END;
```

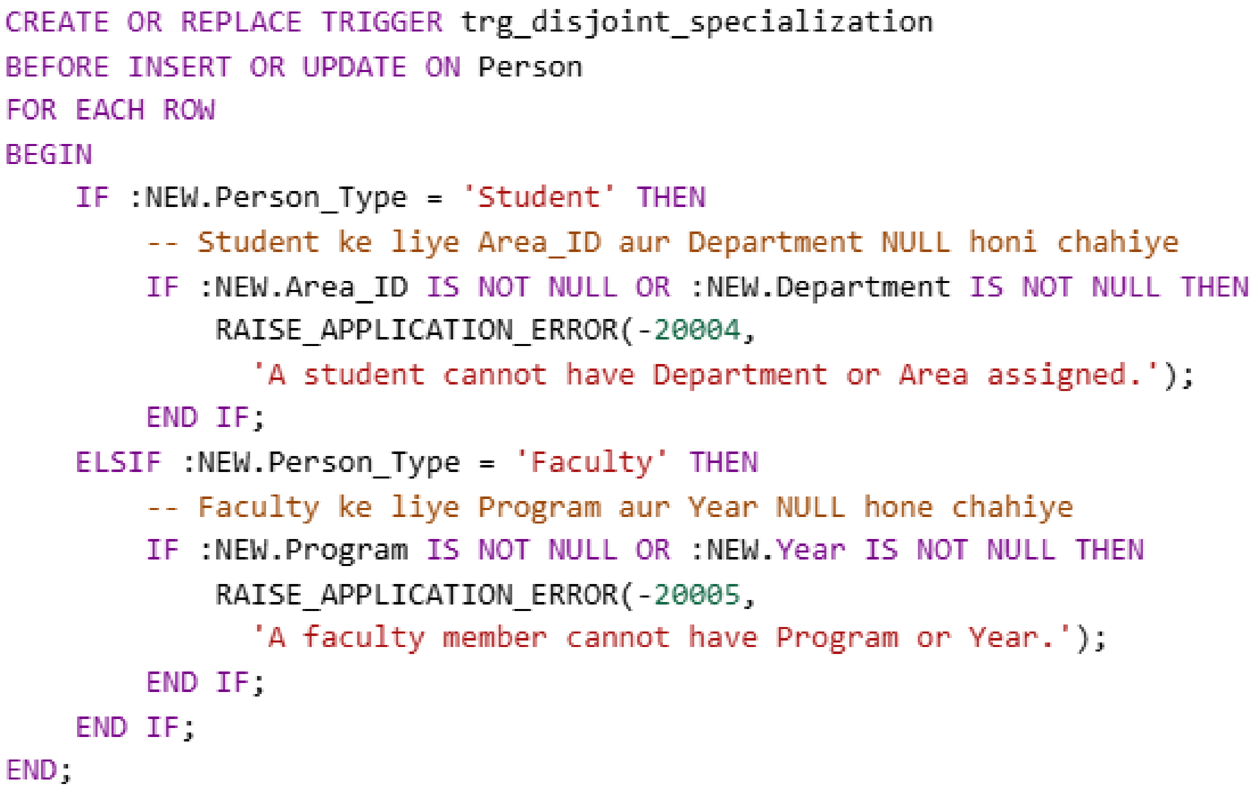
Trigger 3: Conference cannot be marked as Active unless at least one participant is linked



```
CREATE OR REPLACE TRIGGER trg_active_conf_participants
BEFORE INSERT OR UPDATE ON Conference
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    IF :NEW.Conference_Status = 'Active' THEN
        SELECT COUNT(*)
        INTO v_count
        FROM Conference_Participation
        WHERE Conference_ID = :NEW.Conference_ID;

        IF v_count = 0 THEN
            RAISE_APPLICATION_ERROR(-20003,
              'Conference cannot be marked as Active unless at least one participant is linked.');
        END IF;
    END IF;
END;
```

Trigger 4: Person can only be either Student or Faculty (disjoint specialization)



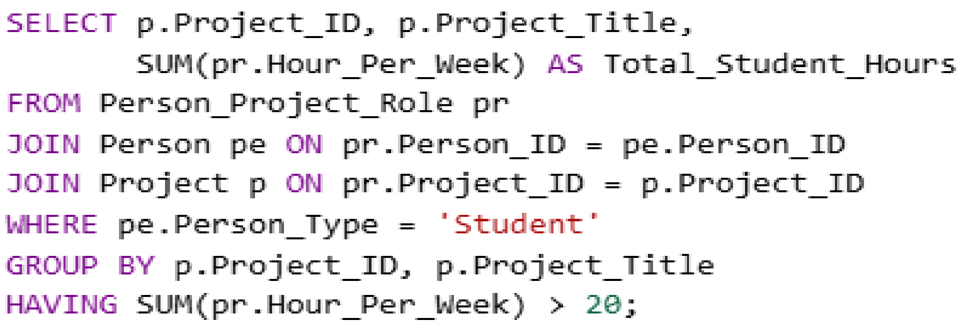
```
CREATE OR REPLACE TRIGGER trg_disjoint_specialization
BEFORE INSERT OR UPDATE ON Person
FOR EACH ROW
BEGIN
     IF :NEW.Person_Type = 'Student' THEN
         -- Student ke liye Area_ID aur Department NULL honi chahiye
         IF :NEW.Area_ID IS NOT NULL OR :NEW.Department IS NOT NULL THEN
             RAISE_APPLICATION_ERROR(-20004,
               'A student cannot have Department or Area assigned.');
         END IF;
     ELSIF :NEW.Person_Type = 'Faculty' THEN
         -- Faculty ke liye Program aur Year NULL hone chahiye
         IF :NEW.Program IS NOT NULL OR :NEW.Year IS NOT NULL THEN
             RAISE_APPLICATION_ERROR(-20005,
               'A faculty member cannot have Program or Year.');
         END IF;
     END IF;
END;
```

Structured Query Language (SQL) – Analytical Queries

The following SQL queries were executed to validate and analyze the Research & Conference Management System database. These queries test relationships, constraints, and advanced data insights from the normalized schema.

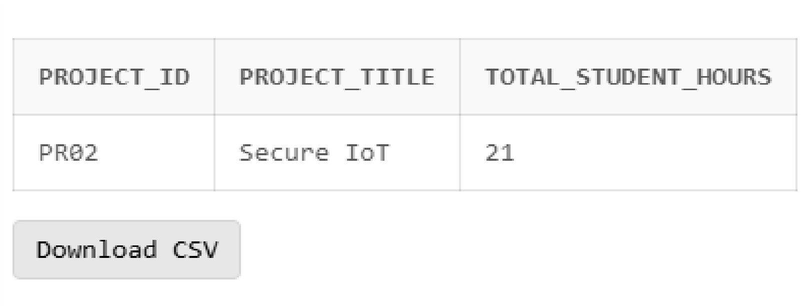
1. List research projects with total student hours/week > 20.

QUERY



```
SELECT p.Project_ID, p.Project_Title,
       SUM(pr.Hour_Per_Week) AS Total_Student_Hours
FROM Person_Project_Role pr
JOIN Person pe ON pr.Person_ID = pe.Person_ID
JOIN Project p ON pr.Project_ID = p.Project_ID
WHERE pe.Person_Type = 'Student'
GROUP BY p.Project_ID, p.Project_Title
HAVING SUM(pr.Hour_Per_Week) > 20;
```

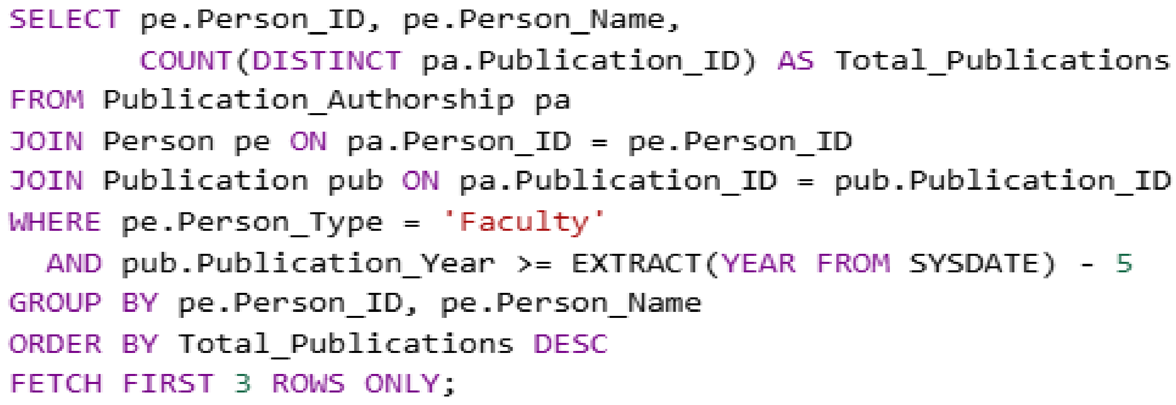
OUTPUT



PROJECT_ID	PROJECT_TITLE	TOTAL_STUDENT_HOURS
PR02	Secure IoT	21

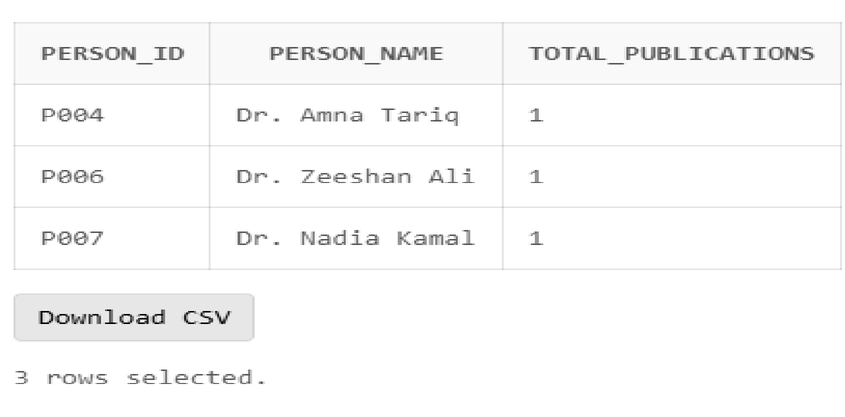
2. Top 3 most published faculty in the last 5 years

QUERY



```
SELECT pe.Person_ID, pe.Person_Name,
       COUNT(DISTINCT pa.Publication_ID) AS Total_Publications
FROM Publication_Authorship pa
JOIN Person pe ON pa.Person_ID = pe.Person_ID
JOIN Publication pub ON pa.Publication_ID = pub.Publication_ID
WHERE pe.Person_Type = 'Faculty'
  AND pub.Publication_Year >= EXTRACT(YEAR FROM SYSDATE) - 5
GROUP BY pe.Person_ID, pe.Person_Name
ORDER BY Total_Publications DESC
FETCH FIRST 3 ROWS ONLY;
```

OUTPUT

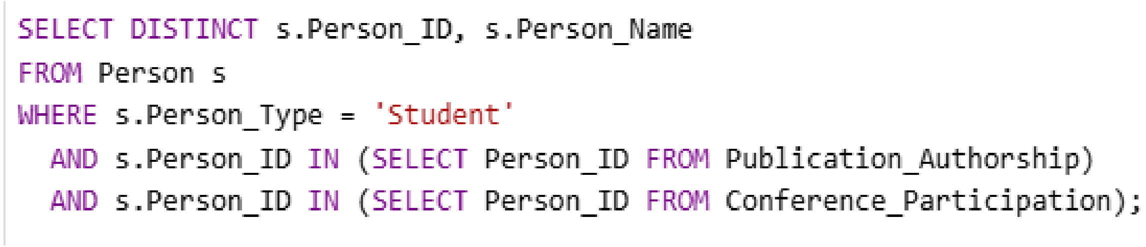


PERSON_ID	PERSON_NAME	TOTAL_PUBLICATIONS
P004	Dr. Amna Tariq	1
P006	Dr. Zeeshan Ali	1
P007	Dr. Nadia Kamal	1

3 rows selected.

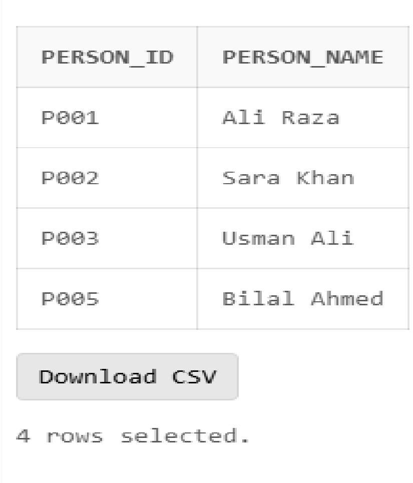
3. Students who have published and attended conferences.

QUERY



```
SELECT DISTINCT s.Person_ID, s.Person_Name
FROM Person s
WHERE s.Person_Type = 'Student'
  AND s.Person_ID IN (SELECT Person_ID FROM Publication_Authorship)
  AND s.Person_ID IN (SELECT Person_ID FROM Conference_Participation);
```

OUTPUT



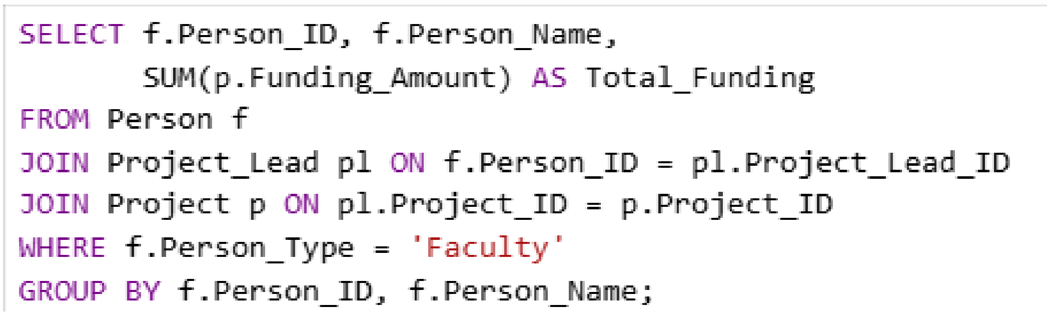
PERSON_ID	PERSON_NAME
P001	Ali Raza
P002	Sara Khan
P003	Usman Ali
P005	Bilal Ahmed

Download CSV

4 rows selected.

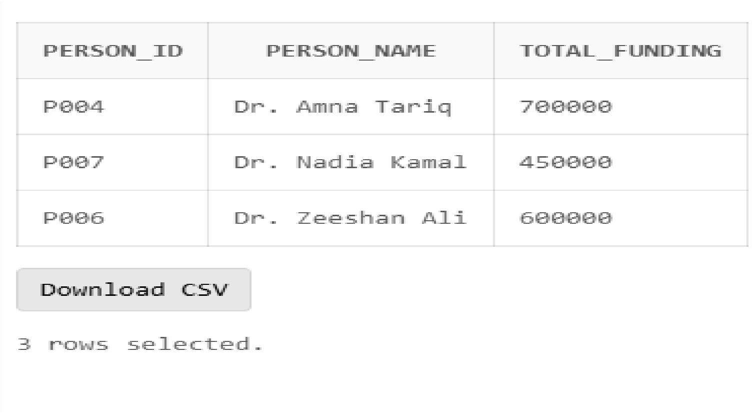
4. Total funding handled by each faculty.

QUERY



```
SELECT f.Person_ID, f.Person_Name,
       SUM(p.Funding_Amount) AS Total_Funding
FROM Person f
JOIN Project_Lead pl ON f.Person_ID = pl.Project_Lead_ID
JOIN Project p ON pl.Project_ID = p.Project_ID
WHERE f.Person_Type = 'Faculty'
GROUP BY f.Person_ID, f.Person_Name;
```

OUTPUT



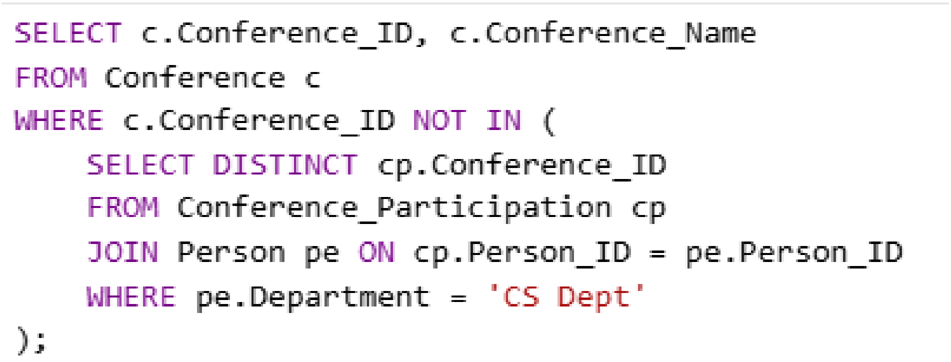
PERSON_ID	PERSON_NAME	TOTAL_FUNDING
P004	Dr. Amna Tariq	700000
P007	Dr. Nadia Kamal	450000
P006	Dr. Zeeshan Ali	600000

Download CSV

3 rows selected.

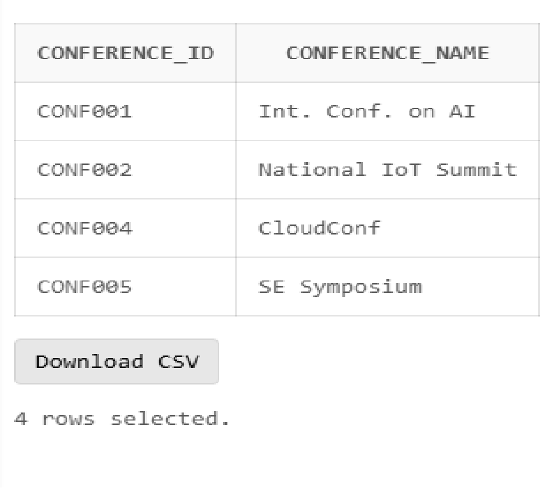
5. Conferences where no CS department member attended

QUERY



```
SELECT c.Conference_ID, c.Conference_Name
FROM Conference c
WHERE c.Conference_ID NOT IN (
    SELECT DISTINCT cp.Conference_ID
    FROM Conference_Participation cp
    JOIN Person pe ON cp.Person_ID = pe.Person_ID
    WHERE pe.Department = 'CS Dept'
);
```

OUTPUT



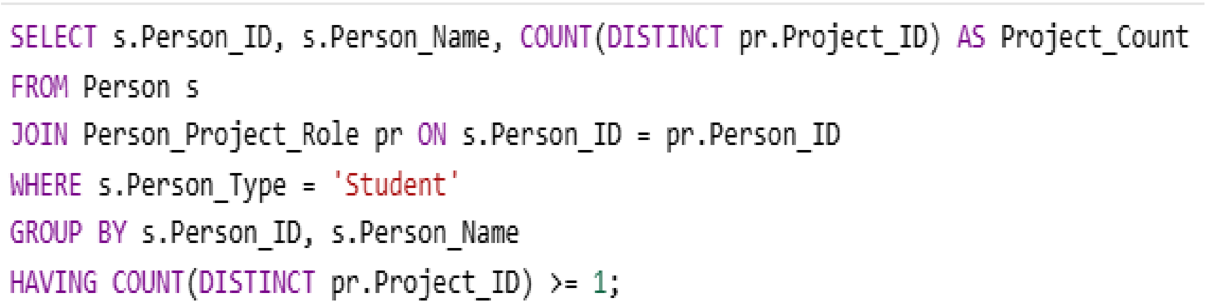
CONFERENCE_ID	CONFERENCE_NAME
CONF001	Int. Conf. on AI
CONF002	National IoT Summit
CONF004	CloudConf
CONF005	SE Symposium

Download CSV

4 rows selected.

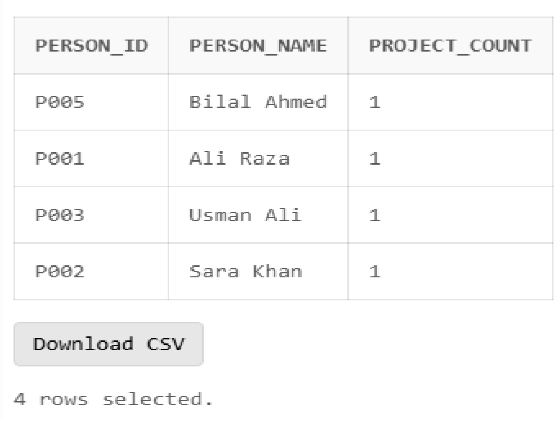
6. List students who are involved in at least 1 project

QUERY



```
SELECT s.Person_ID, s.Person_Name, COUNT(DISTINCT pr.Project_ID) AS Project_Count
FROM Person s
JOIN Person_Project_Role pr ON s.Person_ID = pr.Person_ID
WHERE s.Person_Type = 'Student'
GROUP BY s.Person_ID, s.Person_Name
HAVING COUNT(DISTINCT pr.Project_ID) >= 1;
```

OUTPUT



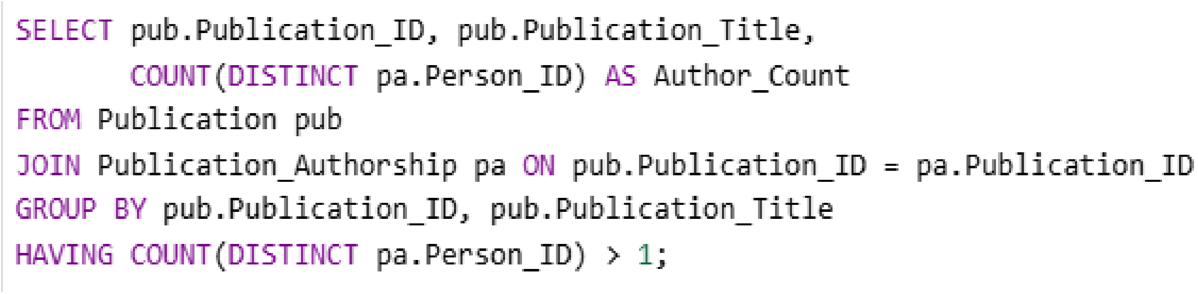
PERSON_ID	PERSON_NAME	PROJECT_COUNT
P005	Bilal Ahmed	1
P001	Ali Raza	1
P003	Usman Ali	1
P002	Sara Khan	1

Download CSV

4 rows selected.

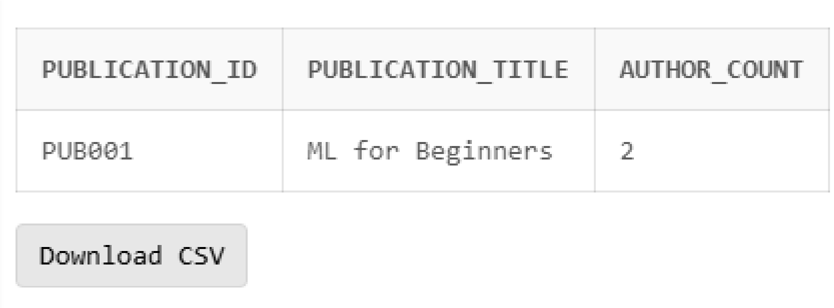
7. Publications that have Publications that have more than one author

QUERY



```
SELECT pub.Publication_ID, pub.Publication_Title,
       COUNT(DISTINCT pa.Person_ID) AS Author_Count
FROM Publication pub
JOIN Publication_Authorship pa ON pub.Publication_ID = pa.Publication_ID
GROUP BY pub.Publication_ID, pub.Publication_Title
HAVING COUNT(DISTINCT pa.Person_ID) > 1;
```

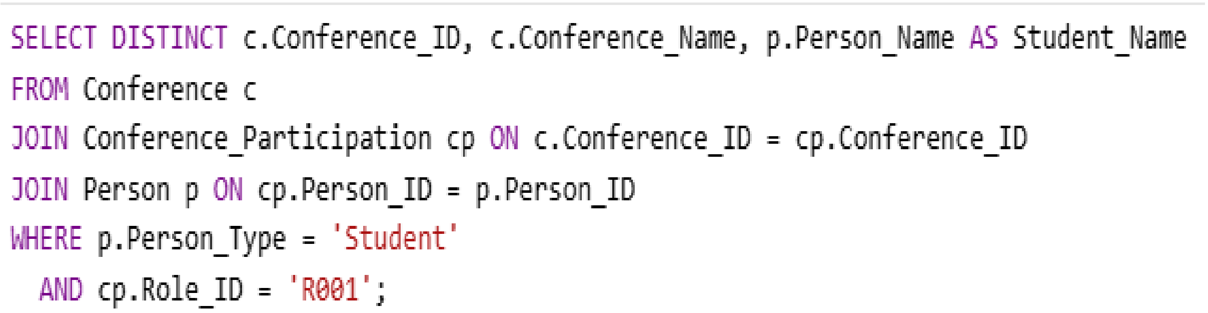
OUTPUT



PUBLICATION_ID	PUBLICATION_TITLE	AUTHOR_COUNT
PUB001	ML for Beginners	2

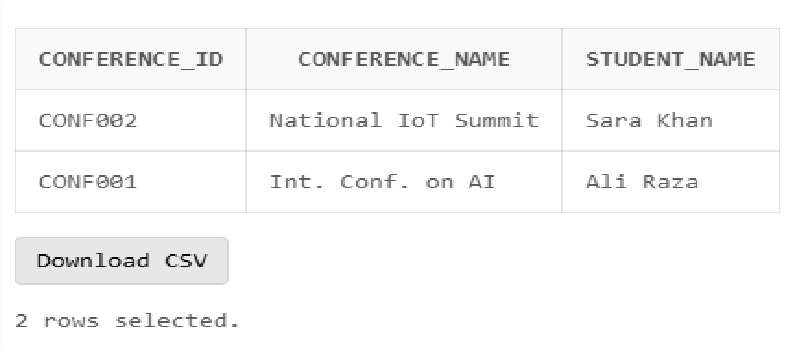
8. Conferences where students presented

QUERY



```
SELECT DISTINCT c.Conference_ID, c.Conference_Name, p.Person_Name AS Student_Name
FROM Conference c
JOIN Conference_Participation cp ON c.Conference_ID = cp.Conference_ID
JOIN Person p ON cp.Person_ID = p.Person_ID
WHERE p.Person_Type = 'Student'
  AND cp.Role_ID = 'R001';
```

OUTPUT

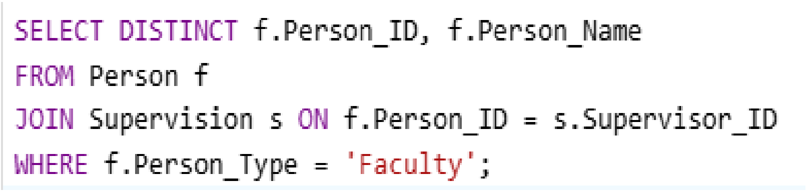


CONFERENCE_ID	CONFERENCE_NAME	STUDENT_NAME
CONF002	National IoT Summit	Sara Khan
CONF001	Int. Conf. on AI	Ali Raza

2 rows selected.

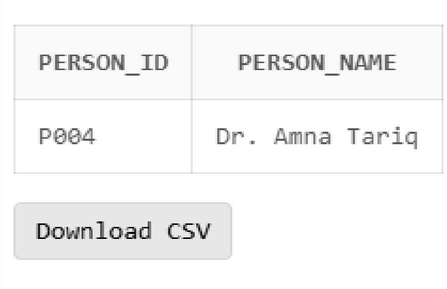
9. List faculty who have supervised at least one student.

QUERY



```
SELECT DISTINCT f.Person_ID, f.Person_Name
FROM Person f
JOIN Supervision s ON f.Person_ID = s.Supervisor_ID
WHERE f.Person_Type = 'Faculty';
```

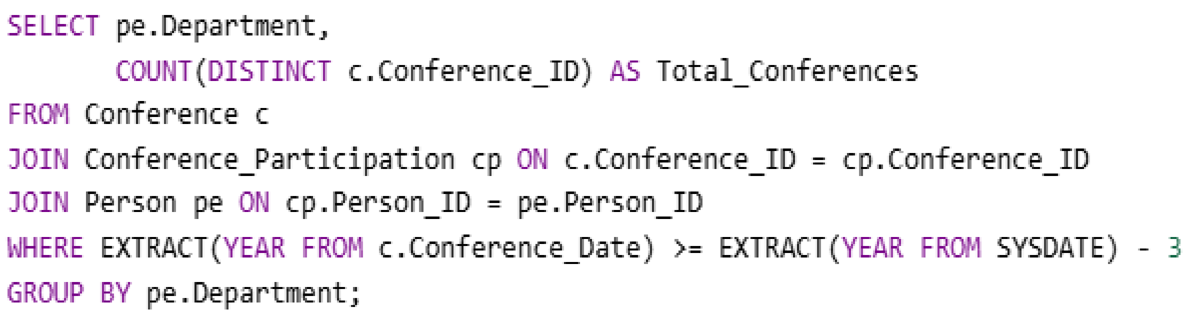
OUTPUT



PERSON_ID	PERSON_NAME
P004	Dr. Amna Tariq

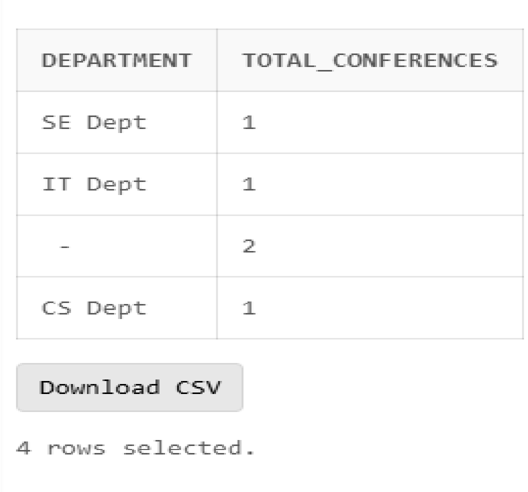
10. Find the total number of conferences each department participated in during the last 3 years.

QUERY



```
SELECT pe.Department,
       COUNT(DISTINCT c.Conference_ID) AS Total_Conferences
FROM Conference c
JOIN Conference_Participation cp ON c.Conference_ID = cp.Conference_ID
JOIN Person pe ON cp.Person_ID = pe.Person_ID
WHERE EXTRACT(YEAR FROM c.Conference_Date) >= EXTRACT(YEAR FROM SYSDATE) - 3
GROUP BY pe.Department;
```

OUTPUT



DEPARTMENT	TOTAL_CONFERENCES
SE Dept	1
IT Dept	1
-	2
CS Dept	1

```
4 rows selected.
```

NoSQL-Based Representation (MongoDB Perspective)

Alternative non-relational structure of the same system

Why NoSQL?

Relational design is perfect for structured data and strict constraints, but research systems often evolve with new attributes and interdisciplinary collaborations. A NoSQL approach (like MongoDB) offers flexibility, horizontal scalability, and stores related information in a single document without complex joins.

Proposed NoSQL Model

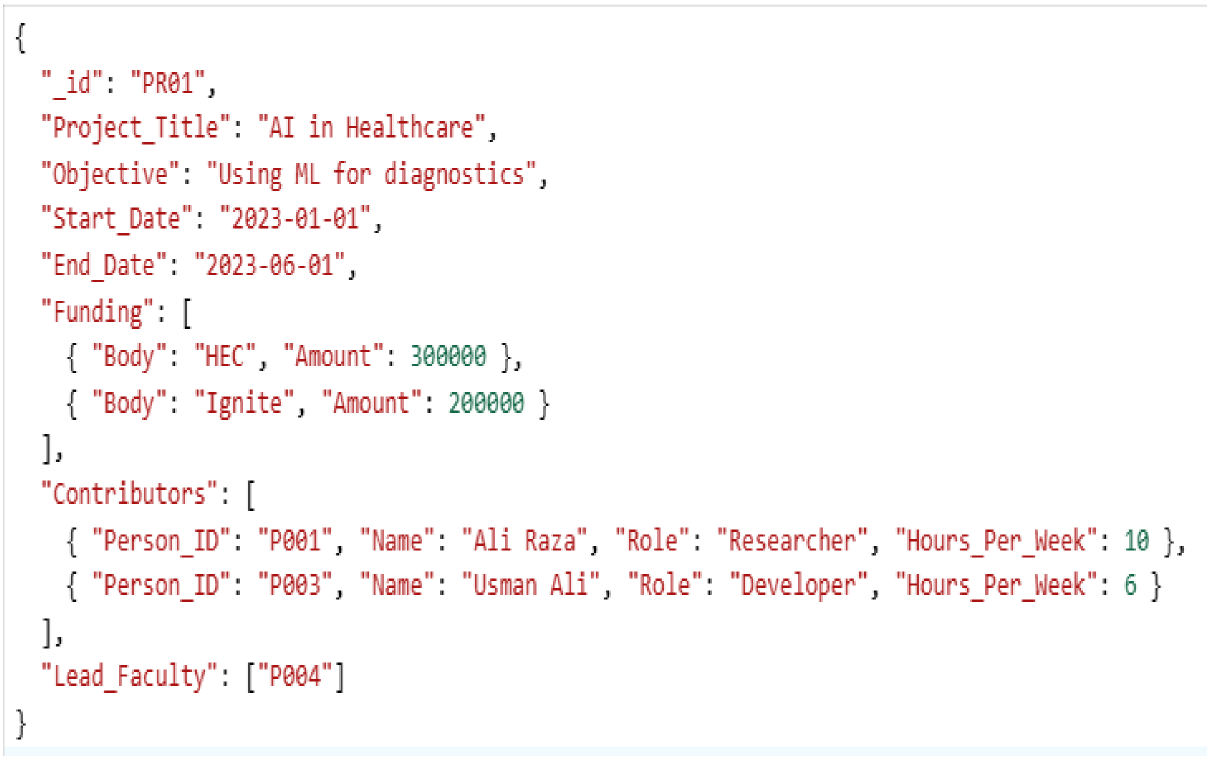
For this system, a document-oriented model is ideal.

Each major entity is stored as a collection (e.g., projects, persons, publications, conferences). Related details are embedded as arrays or sub-documents.

Sample MongoDB Document for a Project

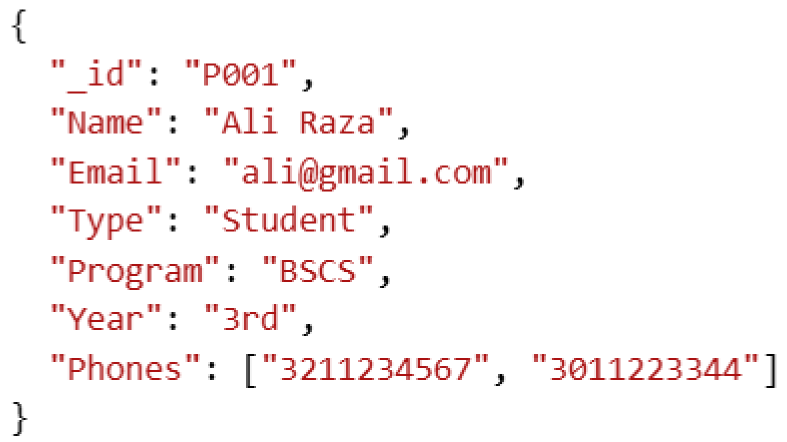
Below is a sample document structure representing a research project in MongoDB:

Example: Project document in MongoDB



```
{
  "_id": "PR01",
  "Project_Title": "AI in Healthcare",
  "Objective": "Using ML for diagnostics",
  "Start_Date": "2023-01-01",
  "End_Date": "2023-06-01",
  "Funding": [
    { "Body": "HEC", "Amount": 300000 },
    { "Body": "Ignite", "Amount": 200000 }
  ],
  "Contributors": [
    { "Person_ID": "P001", "Name": "Ali Raza", "Role": "Researcher", "Hours_Per_Week": 10 },
    { "Person_ID": "P003", "Name": "Usman Ali", "Role": "Developer", "Hours_Per_Week": 6 }
  ],
  "Lead_Faculty": ["P004"]
}
```

Example: Person document in MongoDB’



```
{
  "_id": "P001",
  "Name": "Ali Raza",
  "Email": "ali@gmail.com",
  "Type": "Student",
  "Program": "BSCS",
  "Year": "3rd",
  "Phones": ["3211234567", "3011223344"]
}
```

The above JSON documents illustrate how data is modeled in MongoDB. Instead of using multiple tables and joins, related information—such as contributors, funding, and contact details—is embedded within a single document. This approach offers greater flexibility and scalability as new attributes or nested data are introduced.

Constraints in NoSQL (MongoDB)

MongoDB does not support traditional constraints like foreign keys, joins, or triggers. As a result, rules such as:

- Max two active projects per student
- Each publication must include a faculty author are enforced through application-level logic or aggregation pipelines within the backend.

Benefits of MongoDB

A. Flexible Schema – Easily adapts to changing data structures.
B. Embedded Documents – Stores related data (e.g., authors in publications) in a single document.
C. Optimized for Semi-Structured Data – Ideal for large or evolving datasets.
D. Faster Reads – Minimal joins lead to better performance.

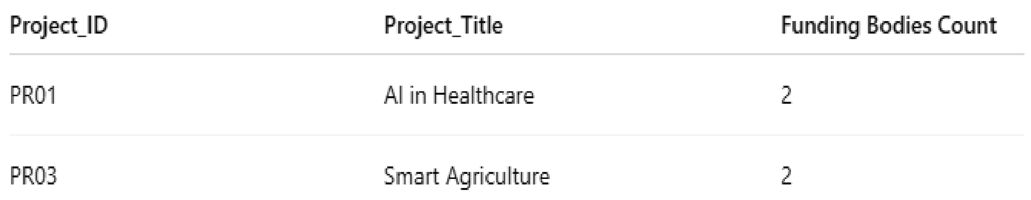
Sample MongoDB Queries

1. Find projects with multiple funding bodies

QUERY

```
db.projects.find({ "Funding.1": { $exists: true } })
```

OUTPUT



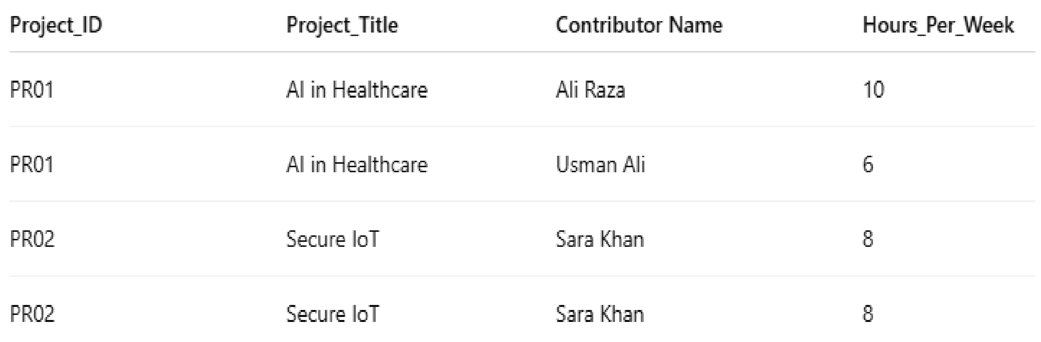
Project_ID	Project_Title	Funding Bodies Count
PR01	AI in Healthcare	2
PR03	Smart Agriculture	2

2.Find students involved in projects with >5 hours/week

QUERY

```
db.projects.find({ "Contributors.Hours_Per_Week": { $gt: 5 } })
```

OUTPUT



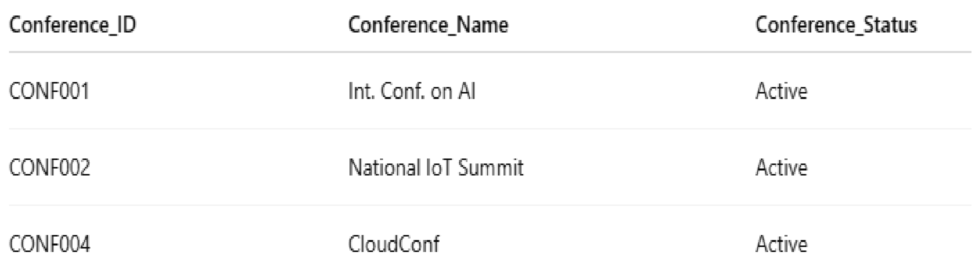
Project_ID	Project_Title	Contributor Name	Hours_Per_Week
PR01	AI in Healthcare	Ali Raza	10
PR01	AI in Healthcare	Usman Ali	6
PR02	Secure IoT	Sara Khan	8
PR02	Secure IoT	Sara Khan	8

3.Find conferences marked Active

QUERY

```
db.conferences.find({ "Conference_Status": "Active" })
```

OUTPUT



Conference_ID	Conference_Name	Conference_Status
CONF001	Int. Conf. on AI	Active
CONF002	National IoT Summit	Active
CONF004	CloudConf	Active

Conclusion

The University-Research-Conference Management System has been designed using a relational model and fully normalized up to Third Normal Form (3NF). All core academic relationships — such as faculty supervision, student participation in projects, authorship of publications, and conference activities — have been represented accurately using entity relationships and relationship tables.

The model ensures data consistency, eliminates redundancy, and is ready for implementation in any relational database system.

This system does more than store data — it preserves the academic journey of individuals, their contributions, and their collaborative milestones. From supervision to publications and conference participation, each entity plays a role in shaping knowledge. With thoughtful design and normalization, the database ensures that this story remains organized, accurate, and accessible for years to come.