IUPUI
SCHOOL OF INFORMATICS AND COMPUTING

Date: 12/13/2022
Event: Data Book: Virtual Symposium

Mentoring System Database

Data Book: Virtual Symposium

Presentation : Hasaranga Jayathilake
MS Applied Data Science - IUPUI

InformaticsIUPUI @IUPUInformatics soicindy

Today… Mentoring System Database

- What is **mentoring**
- Mentoring system **users** and **their expectations** from the database
- **Objectives** of the **mentoring system database**
- **Used data types**
- Data **included** and their **purpose**
- Database entity relationship diagram (**ERD**) and **objectives**
- Reflections on **struggles**/**roadblocks** and **successes**/**accomplishments**
- **Learnings** for **future** projects

What is Mentoring

- Mentoring is a process to help employees to other employees to increase knowledge levels; and it is is considered one method to improve the knowledge level of the employees (Clark, 2017).
- Prime benefits of the mentoring
  - less cost to improve the knowledge level in a quick turnaround time (Orlova, 2021).
  - improving the bond among employees (Orlova, 2021).
  - Familiarization among employees on strengths and weaknesses of each other (Ng, 2022).
  - improving overall productivity and accuracy of the works (Adil et al., 2021).
- Secondary Benefits of the mentoring
  - Helps the organization to be more competitive shape to meet the demands of the customers (Ge et al., 2010; Orlova, 2021).
  - Helps to create a business ecosystem with a sustainable competitive advantage for the organization (Eriksson et al., 2022).

Database Users and Expectations

User	Workplace Duties	User Expectations from Mentoring Database
Management	• **Enhance customer satisfaction** • Reduce the **software development pipeline** to beat the competition • To **achieve sustainable competitive advantage**	To **identify** the **new talented** employees
		To **increase** the **productivity** of the employees
		Understand the **knowledge level** and **competencies** of the employees
Mentors	• **Showcase their talent** for their co-workers and management • To **get more rewards** (incentives, career succession)	To **understand the performance** level and the progress level of the mentees and improve them
		Resolve the issues faced by the **mentees**.
		To **interact** more easily with mentees
		To track the level of the mentoring programs and their effect on their rewards from the organization.
Mentees	• **Develop** as per the **requirements** • **Reduce** the **bugs** and **re-works** of their work • **Increase** their **performance** to increase their benefits	To get **quick help from co-workers**
		To **complete** the **assigned tasks** for them, **accurately**
		To be **updated** with **new technical advancements**
		To **increase networking time** with **their peers**

Objectives of the Mentoring System Database

- Having a mentoring system, will enable to check on Up-to-date
  - progress of the programs
  - effectiveness of the process as feedbacks
  - interaction level among employees
  - motivation towards the programs
  - rewards or incentives for the good preforming mentors (Eriksson et al., 2022).
- Therefore,
  - From this database trying to cover those objectives as

Objectives	Tables Used
Progress	SESSION_COMPLETED
Effectiveness	MENTEEE_FEEDBACK and MENTOR_FEEDBACK
Interaction	CONVERSATION_REGISTY
Motivation	ATTEND_REGISTRY
Rewards	MENTOR_REWARDS

Data types Used with database examples

– Most of the primary keys were developed using the **variable character (VARCHAR) data type** and show the what sort of a character need to be created in terms of the each primary key when it is a VARCHAR data type in the database by adding a comment in the database.

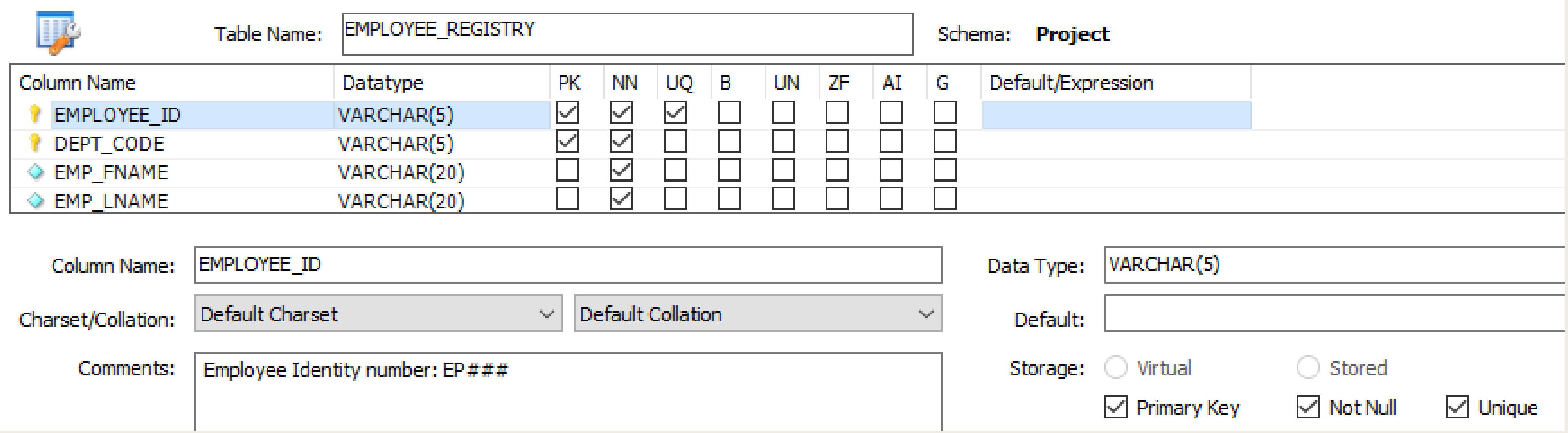


Table Name: EMPLOYEE_REGISTRY Schema: **Project**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
EMPLOYEE_ID	VARCHAR(5)	☑	☑	☑	□	□	□	□	□	
DEPT_CODE	VARCHAR(5)	☑	☑	□	□	□	□	□	□	
EMP_FNAME	VARCHAR(20)	□	☑	□	□	□	□	□	□	
EMP_LNAME	VARCHAR(20)	□	☑	□	□	□	□	□	□	

Column Name: EMPLOYEE_ID

Data Type: VARCHAR(5)

Charset/Collation: Default Charset | Default Collation

Default:

Comments: Employee Identity number: EP###

Storage: ○ Virtual ○ Stored

☑ Primary Key ☑ Not Null ☑ Unique

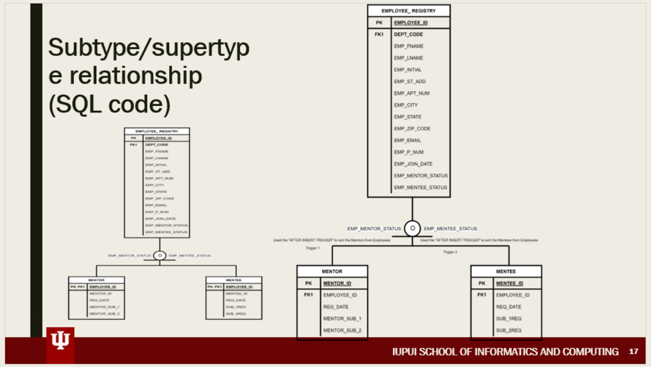
EMPLOYEE_REGISRTY ENTITY (SQL code)

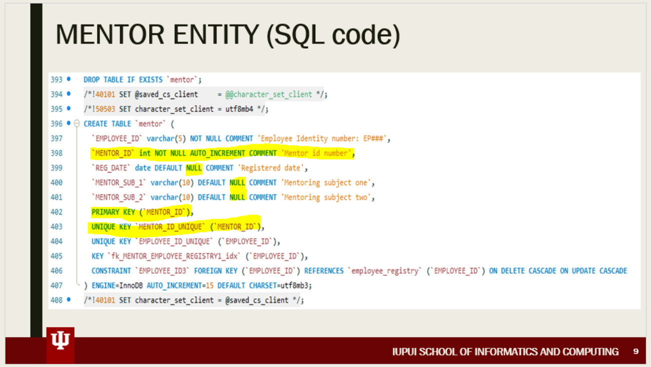
```
DROP TABLE IF EXISTS `employee_registry`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `employee_registry` (
  `EMPLOYEE_ID` varchar(5) NOT NULL COMMENT 'Employee Identity number: EP###',
  `DEPT_CODE` varchar(5) DEFAULT NULL COMMENT 'Department code\nStart from DT##',
  `EMP_FNAME` varchar(20) NOT NULL COMMENT 'Employee first name',
  `EMP_LNAME` varchar(20) NOT NULL COMMENT 'Employee last name',
  `EMP_INTIAL` char(1) NOT NULL COMMENT 'Employee initial',
  `EMP_ST_ADD` varchar(20) NOT NULL COMMENT 'Employee street address',
  `EMP_APT_NUM` varchar(6) DEFAULT NULL COMMENT 'Employee apartment number',
  `EMP_CITY` varchar(20) NOT NULL COMMENT 'Employee city',
  `EMP_STATE` varchar(2) NOT NULL COMMENT 'Employee State',
  `EMP_ZIP_CODE` varchar(5) NOT NULL COMMENT 'Employee zip code',
  `EMP_EMAIL` varchar(45) NOT NULL COMMENT 'Employee personal email',
  `EMP_P_NUM` varchar(10) NOT NULL COMMENT 'Employee personal phone number',
  `EMP_JOIN_DATE` date NOT NULL COMMENT 'Employee joined date',
  `EMP_MENTOR_STATUS` varchar(3) NOT NULL COMMENT 'Employee mentor registration status',
  `EMP_MENTEE_STATUS` varchar(3) NOT NULL COMMENT 'Employee mentee registration status',
  PRIMARY KEY (`EMPLOYEE_ID`),
  UNIQUE KEY `EMPLOYEE_ID_UNIQUE` (`EMPLOYEE_ID`),
  KEY `fk_EMPLOYEE_REGISTRY_ORG_DEPAT1_idx` (`DEPT_CODE`),
  CONSTRAINT `fk_EMPLOYEE_REGISTRY_ORG_DEPAT1` FOREIGN KEY (`DEPT_CODE`) REFERENCES `org_depat` (`DEPT_CODE`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
/*!40101 SET character_set_client = @saved_cs_client */;
```

IUPUI SCHOOL OF INFORMATICS AND COMPUTING 7

Data types Used with database examples Cont…

- Especially, for the **primary keys** of the MENTEE and MENTOR entities used the **integer (INT) data type** with **auto increment function.** That will allow to create the trigger which enable the subtype/supertype relationship;
  - EMPLOYEE_REGISTRY → MENTEE (supertype to subtype relationship)
  - EMPLOYEE_REGISTRY → MENTOR (supertype to subtype relationship)

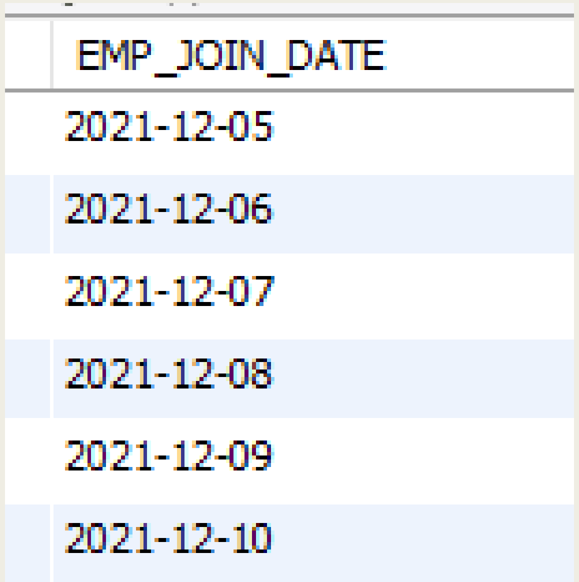

Subtype/supertype relationship (SQL code)


MENTOR ENTITY (SQL code)

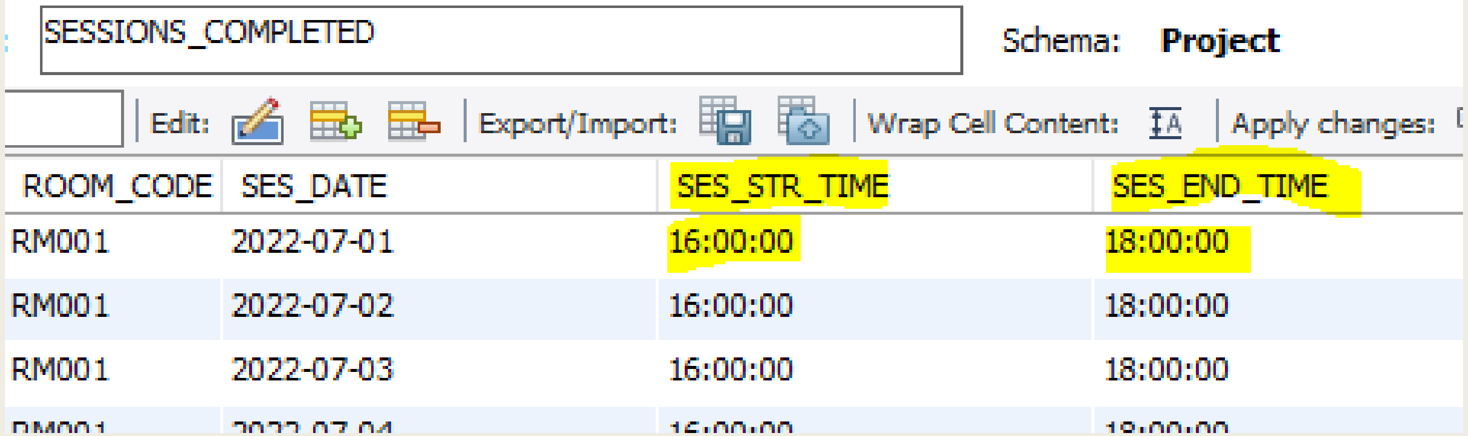
```
DROP TABLE IF EXISTS `mentor`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `mentor` (
  `EMPLOYEE_ID` varchar(5) NOT NULL COMMENT 'Employee Identity number: EP###',
  `MENTOR_ID` int NOT NULL AUTO_INCREMENT COMMENT 'Mentor id number',
  `REG_DATE` date DEFAULT NULL COMMENT 'Registered date',
  `MENTOR_SUB_1` varchar(10) DEFAULT NULL COMMENT 'Mentoring subject one',
  `MENTOR_SUB_2` varchar(10) DEFAULT NULL COMMENT 'Mentoring subject two',
  PRIMARY KEY (`MENTOR_ID`),
  UNIQUE KEY `MENTOR_ID_UNIQUE` (`MENTOR_ID`),
  UNIQUE KEY `EMPLOYEE_ID_UNIQUE` (`EMPLOYEE_ID`),
  KEY `fk_MENTOR_EMPLOYEE_REGISTRY1_idx` (`EMPLOYEE_ID`),
  CONSTRAINT `EMPLOYEE_ID3` FOREIGN KEY (`EMPLOYEE_ID`) REFERENCES `employee_registry` (`EMPLOYEE_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8mb3;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Data types Used with database examples Cont…

- Mainly VARCHAR data type was used in as the data type for the inserting information on the non-primary key attributes
- DATE data type was used on all the date entries though out the database with YYYY-MM-DD
- TIME data type was used on all the data entries though out the database with 24-hour standard time with HH:MM:SS



EMP_JOIN_DATE
2021-12-05
2021-12-06
2021-12-07
2021-12-08
2021-12-09
2021-12-10



SESSIONS_COMPLETED

Schema: **Project**

ROOM_CODE	SES_DATE	SES_STR_TIME	SES_END_TIME
RM001	2022-07-01	16:00:00	18:00:00
RM001	2022-07-02	16:00:00	18:00:00
RM001	2022-07-03	16:00:00	18:00:00
RM001	2022-07-04	16:00:00	18:00:00

Information used in Database

- 22 tables were used in order to develop the database
  - 4 main parent tables
  - 8 child tables
  - 10 connecting tables for ensure the M:N relationships
  - 20 business rules used make Entity relationship model (ERM)
    (Page 26 of Database)
  - 2 assumptions and
    2 constraints were used in developing the ERM

Parent Tables	Child Tables	Connecting Tables
ORG_DEPAT	ROOM_REGISTRY	ORG_DEPAT_ROOM_ASSC
BULD_REGISTRY	EMPLOYEE_ REGISTRY	MENTOR_PROG_ASSC
PROG_CONDUCTED	MENTOR	MENTEE_PROG_ASSC
ATTEND_REGISTRY	MENTEE	METE_FEDBCK_ASSC
	MENTOR_FEEDBACK	MENTEE_FEEDBACK
	MENTOR_REWARDS	MEOR_FEDBCK_ASSC
	SESSIONS_COMPLETED	ATTEND_REG_ASSC
	CONVERSATION_REGISTY	ATTEND_SESSION_ASSC
		CONV_MENTEE_ASSC
		CONV_MENTOR_ASSC

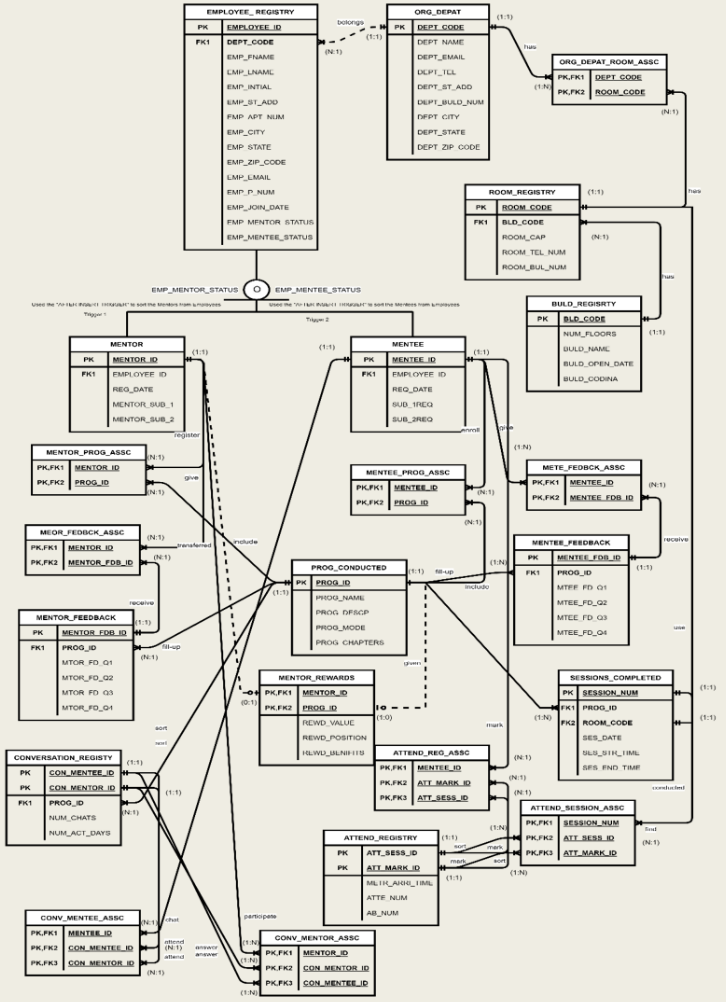
Information used in Database Cont…

- Entity relationship model (ERM) components of the database

Entity	Relationship	Connectivity	Entity
ORG_DEPAT	Belongs	1:M	EMPLOYEE_ REGISTRY
ORG_DEPAT	Has	1:M	ORG_DEPAT_ROOM_ASSC
ROOM_REGISTRY	Has	1:M	ORG_DEPAT_ROOM_ASSC
BULD_REGISRTY	Has	1:M	ROOM_REGISTRY
EMPLOYEE_ REGISTRY	Is a	0:M	MENTEE
EMPLOYEE_ REGISTRY	Is a	0:M	MENTOR
MENTEE	Enroll	1:M	MENTEE_PROG_ASSC
PROG_CONDUCTED	Include	1:M	MENTEE_PROG_ASSC
MENTOR	Register	1:M	MENTOR_PROG_ASSC
PROG_CONDUCTED	Include	1:M	MENTOR_PROG_ASSC
MENTEE	Give	1:M	METE_FEDBCK_ASSC
MENTEEE_FEEDBACK	Receive	1:M	METE_FEDBCK_ASSC
PROG_CONDUCTED	Fill-up	1:M	MENTEEE_FEEDBACK
MENTOR	Give	1:M	MEOR_FEDBCK_ASSC
MENTOR_FEEDBACK	Receive	1:M	MEOR_FEDBCK_ASSC
PROG_CONDUCTED	Fill-up	1:M	MENTOR_FEEDBACK
MENTOR_REWARDS	Given	0:1	PROG_CONDUCTED
MENTOR_REWARDS	Transferred	0:1	MENTOR
SESSION_COMPLETED	Conducted	M:1	PROG_CONDUCTED
ROOM_REGISTRY	Used	1:1	SESSION_COMPLETED
MENTEE	Marked	1:M	ATTEND_REG_ASSC
ATTEND_REGISTRY	Labeled	1:M	ATTEND_REG_ASSC
ATTEND_REGISTRY	Sorted	1:M	ATTEND_SESSION_ASSC
SESSION_COMPLETED	Found	1:M	ATTEND_SESSION_ASSC
CONVERSATION_REGISTY	Happened	M:1	PROG_CONDUCTED
CONVERSATION_REGISTY	Attended	1:M	CONV_MENTEE_ASSC
MENTEE	Chat	1:M	CONV_MENTEE_ASSC
CONVERSATION_REGISTY	Answered	1:M	CONV_MENTOR_ASSC
MENTOR	Participated	1:M	CONV_MENTOR_ASSC

Entity Relationship Diagram of Database

- Using drow.io created the entity relationship diagram.



	EMPLOYEE_ REGISTRY
PK	EMPLOYEE_ID
FK1	DEPT_CODE
	EMP_FNAME
	EMP_LNAME
	EMP_INTIAL
	EMP_ST_ADD
	EMP_APT_NUM
	EMP_CITY
	EMP_STATE
	EMP_ZIP_CODE
	EMP_EMAIL
	EMP_P_NUM
	EMP_JOIN_DATE
	EMP_MENTOR_STATUS
	EMP_MENTEE_STATUS

	ORG_DEPAT
PK	DEPT_CODE
	DEPT_NAME
	DEPT_EMAIL
	DEPT_TEL
	DEPT_ST_ADD
	DEPT_BULD_NUM
	DEPT_CITY
	DEPT_STATE
	DEPT_ZIP_CODE

	ORG_DEPAT_ROOM_ASSC
PK,FK1	DEPT_CODE
PK,FK2	ROOM_CODE

	ROOM_REGISTRY
PK	ROOM_CODE
FK1	BLD_CODE
	ROOM_CAP
	ROOM_TEL_NUM
	ROOM_BUL_NUM

	BULD_REGISRTY
PK	BLD_CODE
	NUM_FLOORS
	BULD_NAME
	BULD_OPEN_DATE
	BULD_CODINA

	MENTOR
PK	MENTOR_ID
FK1	EMPLOYEE_ID
	REG_DATE
	MENTOR_SUB_1
	MENTOR_SUB_2

	MENTEE
PK	MENTEE_ID
FK1	EMPLOYEE_ID
	REQ_DATE
	SUB_1REQ
	SUB_2REQ

	MENTOR_PROG_ASSC
PK,FK1	MENTOR_ID
PK,FK2	PROG_ID

	MENTEE_PROG_ASSC
PK,FK1	MENTEE_ID
PK,FK2	PROG_ID

	METE_FEDBCK_ASSC
PK,FK1	MENTEE_ID
PK,FK2	MENTEE_FDB_ID

	MEOR_FEDBCK_ASSC
PK,FK1	MENTOR_ID
PK,FK2	MENTOR_FDB_ID

	PROG_CONDUCTED
PK	PROG_ID
	PROG_NAME
	PROG_DESCP
	PROG_MODE
	PROG_CHAPTERS

	MENTEE_FEEDBACK
PK	MENTEE_FDB_ID
FK1	PROG_ID
	MTEE_FD_Q1
	MTEE_FD_Q2
	MTEE_FD_Q3
	MTEE_FD_Q4

	MENTOR_FEEDBACK
PK	MENTOR_FDB_ID
FK1	PROG_ID
	MTOR_FD_Q1
	MTOR_FD_Q2
	MTOR_FD_Q3
	MTOR_FD_Q4

	MENTOR_REWARDS
PK,FK1	MENTOR_ID
PK,FK2	PROG_ID
	REWD_VALUE
	REWD_POSITION
	REWD_BENIFITS

	SESSIONS_COMPLETED
PK	SESSION_NUM
FK1	PROG_ID
FK2	ROOM_CODE
	SES_DATE
	SES_STR_TIME
	SES_END_TIME

	CONVERSATION_REGISTY
PK	CON_MENTEE_ID
PK	CON_MENTOR_ID
FK1	PROG_ID
	NUM_CHATS
	NUM_ACT_DAYS

	ATTEND_REG_ASSC
PK,FK1	MENTEE_ID
PK,FK2	ATT_MARK_ID
PK,FK3	ATT_SESS_ID

	ATTEND_SESSION_ASSC
PK,FK1	SESSION_NUM
PK,FK2	ATT_SESS_ID
PK,FK3	ATT_MARK_ID

	ATTEND_REGISTRY
PK	ATT_SESS_ID
PK	ATT_MARK_ID
	METR_ARRI_TIME
	ATTE_NUM
	AB_NUM

	CONV_MENTEE_ASSC
PK,FK1	MENTEE_ID
PK,FK2	CON_MENTEE_ID
PK,FK3	CON_MENTOR_ID

	CONV_MENTOR_ASSC
PK,FK1	MENTOR_ID
PK,FK2	CON_MENTOR_ID
PK,FK3	CON_MENTEE_ID

Used the "AFTER INSERT TRIGGER" to sort the Mentors from Employees
Trigger 1

Used the "AFTER INSERT TRIGGER" to sort the Mentees from Employees
Trigger 2

Reflections on struggles and successes

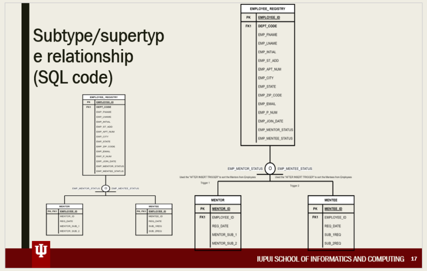
Struggles

- To ensure the proper formation of the foreign key relationship in the database
  - As error, beginning have been received the error on foreign key duplication error.
  - Have been able to identified that need to make the unique foreign key names on each foreign key relationship instant from each other.
  - Moreover, identified the accurate way of making the foreign key relationship is utilization of MYSQL Workbench diagram for;
    - Creating and linking the relationship between different entities of the database would gives the most accurate way to create the foreign keys relationship
    - And, to make the M:N relationship with connecting tables in an accurate manner.

Reflections on struggles and successes Cont…

Struggles

- To build the relationship of subtype/supertype on the database
  - Due to not recognition of the subtype/supertype relationship between entities of EMPLOYEE_REGISTRY, MENTEE and MENTOR.
    - On the first design of the database, EMPLOYEE_REGISTY.EMPLOYEE_ID attribute data must enter manually into the MENTEE and MENTOR entities in a separate time.
      - It would create the data anomalies by having missing data or inaccurate data
      - Even, create more time on data entering the system due to the duplication of work



Subtype/supertype relationship (SQL code)

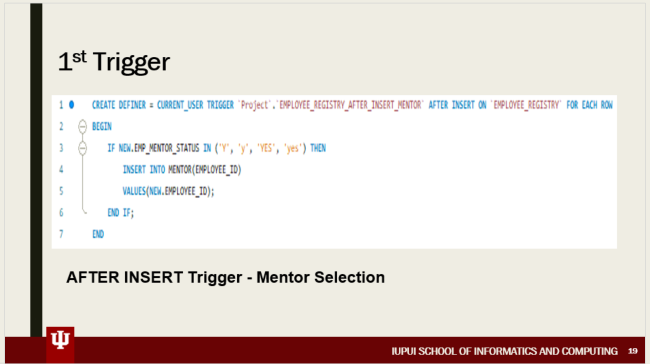
Reflections on struggles and successes Cont…

Struggles

- To build the relationship of subtype/supertype on the database Cont…
  - **To make the proper subtype/supertype relationship**
  - Use 2 triggers of **"After insert trigger"** trigger type in the MYSQL
  - 1st trigger to select the employee data into the MENTOR table from EMPLOYEE_REGISRTY table
  - 2nd trigger to select the employee data into the MENTEE table from EMPLOYEE_REGISRTY table

[MySQL :: MySQL 8.0 Reference Manual :: 25.3.1 Trigger Syntax and Examples]

[MySQL AFTER INSERT Trigger By Practical Examples (mysqltutorial.org)]

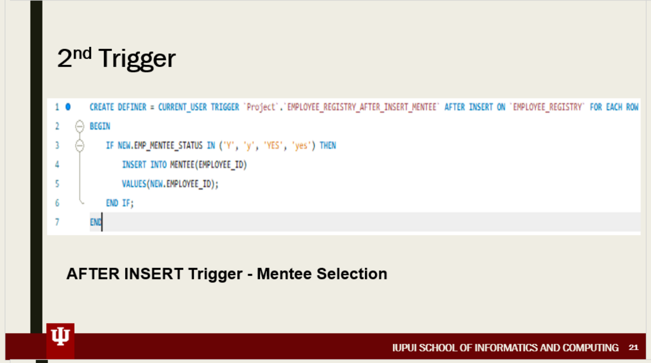


1st Trigger

```sql
CREATE DEFINER = CURRENT_USER TRIGGER `Project`.`EMPLOYEE_REGISTRY_AFTER_INSERT_MENTOR` AFTER INSERT ON `EMPLOYEE_REGISTRY` FOR EACH ROW
BEGIN
    IF NEW.EMP_MENTOR_STATUS IN ('Y', 'y', 'YES', 'yes') THEN
        INSERT INTO MENTOR(EMPLOYEE_ID)
        VALUES(NEW.EMPLOYEE_ID);
    END IF;
END
```

AFTER INSERT Trigger - Mentor Selection

IUPUI SCHOOL OF INFORMATICS AND COMPUTING 19

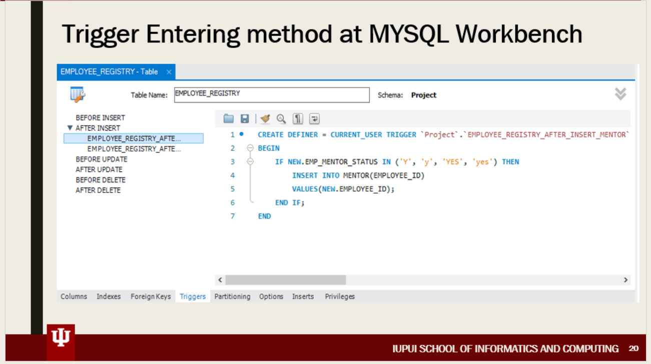


2nd Trigger

```sql
CREATE DEFINER = CURRENT_USER TRIGGER `Project`.`EMPLOYEE_REGISTRY_AFTER_INSERT_MENTEE` AFTER INSERT ON `EMPLOYEE_REGISTRY` FOR EACH ROW
BEGIN
    IF NEW.EMP_MENTEE_STATUS IN ('Y', 'y', 'YES', 'yes') THEN
        INSERT INTO MENTEE(EMPLOYEE_ID)
        VALUES(NEW.EMPLOYEE_ID);
    END IF;
END
```

AFTER INSERT Trigger - Mentee Selection

IUPUI SCHOOL OF INFORMATICS AND COMPUTING 21



Trigger Entering method at MYSQL Workbench

Table Name: EMPLOYEE_REGISTRY

Schema: **Project**

```sql
CREATE DEFINER = CURRENT_USER TRIGGER `Project`.`EMPLOYEE_REGISTRY_AFTER_INSERT_MENTOR`
BEGIN
    IF NEW.EMP_MENTOR_STATUS IN ('Y', 'y', 'YES', 'yes') THEN
        INSERT INTO MENTOR(EMPLOYEE_ID)
        VALUES(NEW.EMPLOYEE_ID);
    END IF;
END
```

IUPUI SCHOOL OF INFORMATICS AND COMPUTING 20

Learnings for future projects

- Effectively usage of MYSQL Workbench for the database planning phase

- To speed-up the database creation process
  - Workbench is helpful to **reduce the coding time** on the basic functions like data entering or creation on the basic different types formats of the database
    - Creating relationships or making connecting tables since Workbench is automated on those functions.
- To enhance the accuracy on building relationships
  - Automated function on creation on the foreign keys enable to improve the errors on the database
- ‘Eliminate or reduce the bugs or data animalities in the database
  - **Lively checking** on the **data types** that **inputting to the database** and indicating those are correct or not

Learnings for future projects

❖ Importance of the proper documentation

- Make it easy for the quick understanding on bugs or errors that are coming through out the database lifecycle.
  - Easy to **find the reason** for the error that is come-up when progress through the creating of the database, inputting data into the database or quiring though the database.
  - Since enabling the **tracing back opportunity**; what went wrong
- Improve the updating opportunities in easily manner for the existing database in future.
  - Ability to **add more functions** or tables as per the **future requirements smoothly**.

Summary

- Understand about the importance mentoring for the organization
- Identified impact from having a database for the mentoring to improve the outcome of the process.
- Examined on the basic requirements and the users of the mentoring system database to have
- Explained on the basic database structure using ERD on the Mentoring system database to have in order to achieve main objectives from that process.
- Examined the straggles that faced during the database creation process
- Reflect on the methods that can be used to enhance the future database projects.

Reference

- Adil, A., Kausar, S., Ameer, S., Ghayas, S., & Shujja, S. (2021). Impact of organizational socialization on organizational citizenship behavior: mediating role of knowledge sharing and role clarity. Current Psychology 2021, 1–9. https://doi.org/10.1007/S12144-021-01899-X

- Clark, K. R. (2017). Managing Multiple Generations in the Workplace. Radiologic Technology, 88(4), 379–396. http://www.radiologictechnology.org/content/88/4/379.full

- Coronel, C., & Morris, S. (2017). DATABASE SYSTEMS. www.cengage.com/highered

- Eriksson, T., Näppä, A., & Robertson, J. (2022). Crafting a paying-it-forward mindset in business: Five principles for a competitive employer branding advantage. Business Horizons. https://doi.org/10.1016/J.BUSHOR.2022.01.001

- Ge, J., Su, X., & Zhou, Y. (2010). Organizational socialization, organizational identification and organizational citizenship behavior: An empirical research of Chinese high-tech manufacturing enterprises. Nankai Business Review, International, 1(2), 166–179. https://doi.org/10.1108/20408741011052573

- Oracle. (2022). MySQL :: MySQL 8.0 Reference Manual :: 25.3.1 Trigger Syntax and Examples. https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html

- Ng, K. Y. N. (2022). Effects of organizational culture, affective commitment and trust on knowledge-sharing tendency. Journal of Knowledge Management, ahead-of-print(ahead-of-print). https://doi.org/10.1108/JKM-03-2022-0191/FULL/PDF

- Orlova, E. v. (2021). Design of Personal Trajectories for Employees' Professional Development in the Knowledge Society under Industry 5.0. Social Sciences 2021, Vol. 10, Page 427, 10(11), 427. https://doi.org/10.3390/SOCSCI10110427

Thank You