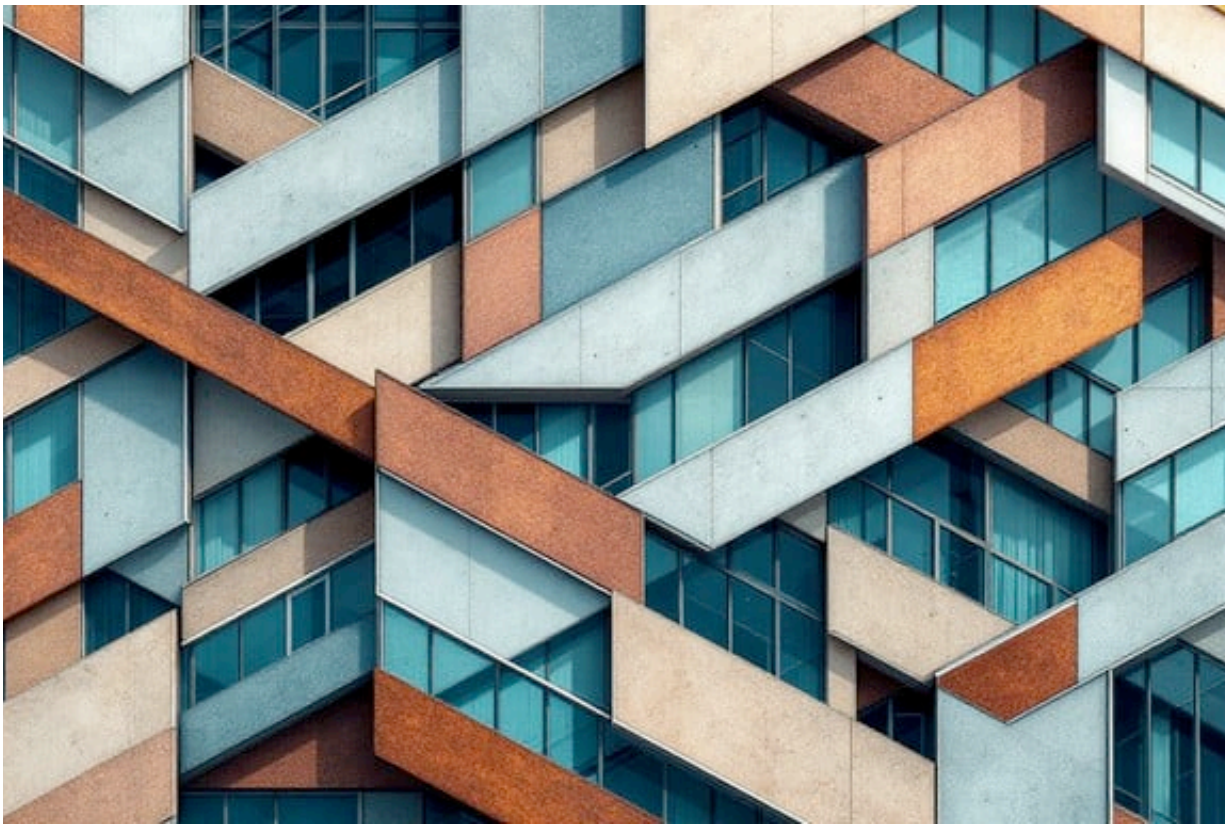
# Database Design Report

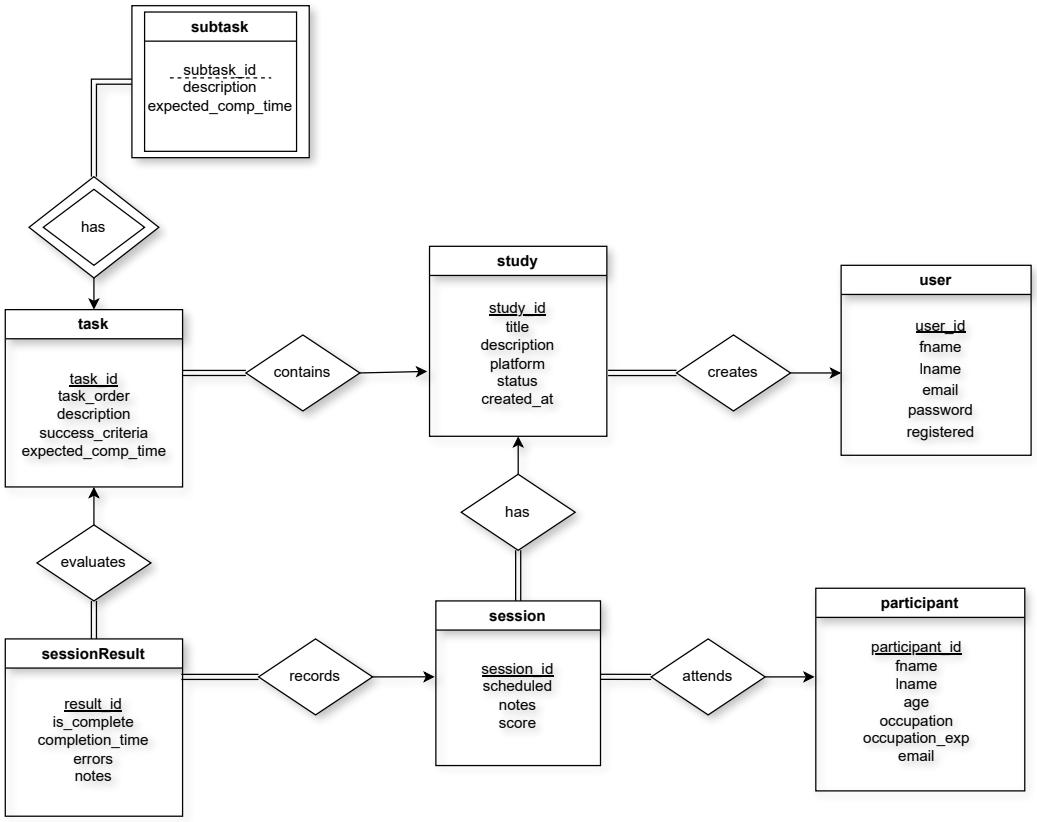
## Database Management Systems



Group 22

November 20, 2025

# Entity-Relationship Diagram



# Entity-Relationship Table Conversion

user(user_id, fname, lname, email, password, registered)
pk: user_id
unique: email

study(study_id, user_id, title, description, platform, status, created_at)
pk: study_id
fk(user_id) references user(user_id)
unique: title
not null: user_id

participant(participant_id, fname, lname, age, occupation, occupation_exp, email)
pk: participant_id

task(task_id, study_id, task_order, description, success_criteria, expected_comp_time)
pk: task_id
fk(study_id) references study(study_id)
unique: (study_id, task_order)
not null: study_id

subtask(task_id, subtask_id, description, expected_comp_time)
pk: (task_id, subtask_id)
fk(task_id) references task(task_id)
not null: task_id

session(session_id, participant_id, study_id, scheduled, notes, score)
pk: session_id
fk(participant_id) references participant(participant_id)
fk(study_id) references study(study_id)
not null: (participant_id, study_id)

sessionResult(result_id, session_id, task_id, is_completed, completion_time, errors, notes)
pk: result_id
fk(session_id) references session(session_id)
fk(task_id) references task(task_id)
ut null: (session_id, task_id)
unique: (task_id, session_id)

# Wide Table

(
    user_id,
    user_fname,
    user_lname,
    user_email,
    user_password,
    user_registered,

    study_id,
    study_title,
    study_description,
    study_platform,
    study_status,
    study_created_at,

    participant_id,
    participant_fname,
    participant_lname,
    participant_age,
    participant_occupation,
    participant_occupation_exp,
    participant_email,

    task_id,
    task_order,
    task_description,
    task_success_criteria,
    task_expected_comp_time,

    subtask_id,
    subtask_description,
    subtask_expected_comp_time,

    session_id,
    session_scheduled,
    session_notes,
    session_score,

    result_id,
    result_is_completed,
    result_completion_time,
    result_errors,
    result_notes
)

The global key for the wide table is **(session_id, task_id)**.

# Functional Dependencies

user_id -> user_fname, user_lname, user_email, user_password, user_registered

user_email -> user_id

study_id -> study_title, study_description, study_platform, study_status, study_created_at, user_id

participant_id -> participant_fname, participant_lname, participant_age, participant_occupation, participant_occupation_exp, participant_email
participant_email -> participant_id

task_id -> study_id, task_order, task_description, task_success_criteria, task_expected_comp_time

study_id, task_order -> task_id

task_id, subtask_id -> subtask_description, subtask_expected_comp_time

session_id -> participant_id, study_id, session_scheduled, session_notes, session_score

result_id -> session_id, task_id, result_completed, result_comp_time, result_errors, result_notes

session_id, task_id -> result_id

# Key

UI -> UF, UL, UE, UP, UR

UE -> UI

SI -> ST, SD, SP, SS, SC, UI

PI -> PF, PL, PA, PO, PX, PE

PE -> PI

TI -> SI, TO, TD, TS, TE

SI, TO -> TI

TI, tI -> tD, tE

sI -> PI, SI, sS, sN, sSC

RI -> sI, TI, RC, RT, RE, RN

sI, TI -> RI

# Minimal Basis

UI -> UF
UI -> UL
UI -> UE
UI -> UP
UI -> UR

UE -> UI

SI -> ST
SI -> SD
SI -> SP
SI -> SS
SI -> SC
SI -> UI

PI -> PF
PI -> PL
PI -> PA
PI -> PO
PI -> PX
PI -> PE

PE -> PI

TI -> SI
TI -> TO
TI -> TD
TI -> TS
TI -> TE

SI, TO -> TI

TI, tI -> tD
TI, tI -> tE

sI -> PI
sI ->  SI
sI -> sS
sI -> sN
sI -> sSC

RI -> sI
RI -> TI
RI -> RC
RI -> RT
RI -> RE
RI -> RN

sI, TI -> RI

# 3NF Synthesis

We use 3NF synthesis rather than BCNF decomposition as it preserves all functional dependencies and results in a practical schema for our application. BCNF, while stricter, would break functional dependency preservation and create additional joins that would not provide any real benefit for our domain.

## Table Creation

R_USER(UI, UF, UL, UE, UP, UR)
R_USER_EMAIL(UE, UI)
R_STUDY(SI, ST, SD, SP, SS, SC, UI)
R_PARTICIPANT(PI, PF, PL, PA, PO, PX, PE)
R_PARTICIPANT_EMAIL(PE, PI)
R_TASK(TI, SI, TO, TD, TS, TE)
R_TASK_IDENTIFICATION(SI, TO, TI)
R_SUBTASK(TI, tI, tD, tE)
R_SESSION(sI, PI, SI, sS, sN, sSC)
R_RESULT(RI, sI, TI, RC, RT, RE, RN)
R_RESULT_IDENTIFICATION(sI, TI, RI)

## Table Redundancy Removal

R_USER(UI, UF, UL, UE, UP, UR)
R_STUDY(SI, ST, SD, SP, SS, SC, UI)
R_PARTICIPANT(PI, PF, PL, PA, PO, PX, PE)
R_TASK(TI, SI, TO, TD, TS, TE)
R_SUBTASK(TI, tI, tD, tE)
R_SESSION(sI, PI, SI, sS, sN, sSC)
R_RESULT(RI, sI, TI, RC, RT, RE, RN)

The global key for this set of tables is **(session_id, task_id)**. The global key can be found in R_RESULT.

# Final 3NF Synthesis Tables

R_USER(UI, UF, UL, UE, UP, UR)
R_STUDY(SI, ST, SD, SP, SS, SC, UI)
R_PARTICIPANT(PI, PF, PL, PA, PO, PX, PE)
R_TASK(TI, SI, TO, TD, TS, TE)
R_SUBTASK(TI, tI, tD, tE)
R_SESSION(sI, PI, SI, sS, sN, sSC)
R_RESULT(RI, sI, TI, RC, RT, RE, RN)

**R_USER**(user_id, user_fname, user_lname, user_email, user_password, user_registered)

**R_STUDY**(study_id, study_title, study_description, study_platform, study_status, study_created_at, user_id)

**R_PARTICIPANT**(participant_id, participant_fname, participant_lname, participant_age, participant_occupation, participant_occupation_exp, participant_email)

**R_TASK**(task_id, study_id, task_order, task_description, task_success_criteria, task_expected_comp_time)

**R_SUBTASK**(task_id, subtask_id, subtask_description, subtask_expected_comp_time)

**R_SESSION**(session_id, participant_id, study_id, session_scheduled, session_notes, session_score)

**R_RESULT(**result_id, session_id, task_id, result_completed, result_comp_time, result_errors, result_notes)

****There is no difference between the initial tables discerned from ER table conversion and 3NF synthesis normalization.*