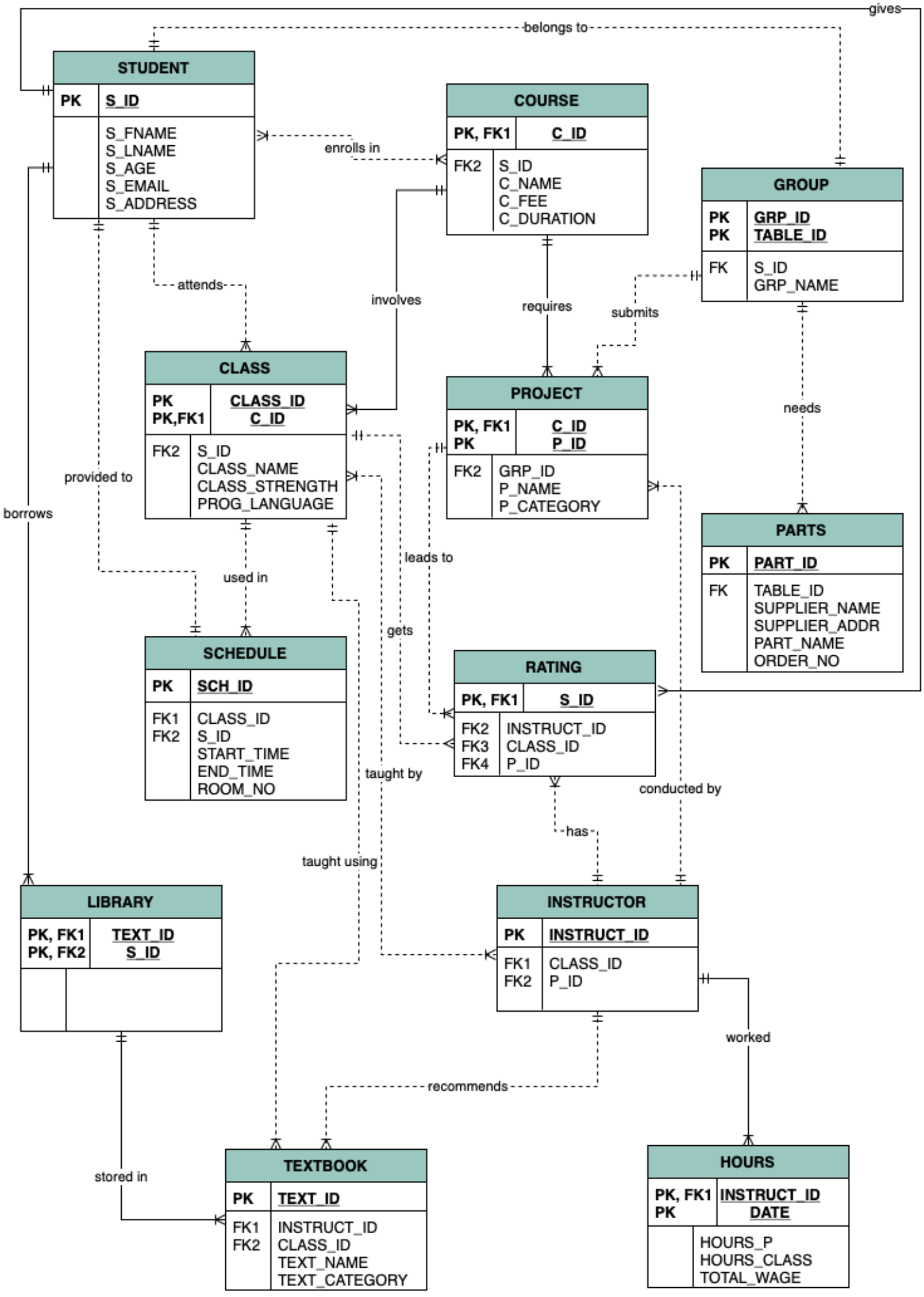
gives
belongs to
STUDENT
PK S_ID
S_FNAME
S_LNAME
S_AGE
S_EMAIL
S_ADDRESS
enrolls in
COURSE
PK, FK1 C_ID
FK2 S_ID
C_NAME
C_FEE
C_DURATION
GROUP
PK GRP_ID
PK TABLE_ID
FK S_ID
GRP_NAME
attends
involves
requires
submits
needs
CLASS
PK CLASS_ID
PK,FK1 C_ID
FK2 S_ID
CLASS_NAME
CLASS_STRENGTH
PROG_LANGUAGE
PROJECT
PK, FK1 C_ID
PK P_ID
FK2 GRP_ID
P_NAME
P_CATEGORY
PARTS
PK PART_ID
FK TABLE_ID
SUPPLIER_NAME
SUPPLIER_ADDR
PART_NAME
ORDER_NO
provided to
borrows
used in
leads to
gets
SCHEDULE
PK SCH_ID
FK1 CLASS_ID
FK2 S_ID
START_TIME
END_TIME
ROOM_NO
RATING
PK, FK1 S_ID
FK2 INSTRUCT_ID
FK3 CLASS_ID
FK4 P_ID
taught by
conducted by
has
taught using
LIBRARY
PK, FK1 TEXT_ID
PK, FK2 S_ID
INSTRUCTOR
PK INSTRUCT_ID
FK1 CLASS_ID
FK2 P_ID
worked
recommends
stored in
TEXTBOOK
PK TEXT_ID
FK1 INSTRUCT_ID
FK2 CLASS_ID
TEXT_NAME
TEXT_CATEGORY
HOURS
PK, FK1 INSTRUCT_ID
PK DATE
HOURS_P
HOURS_CLASS
TOTAL_WAGE

DESIGN ASSUMPTIONS –

STUDENT entity tends to be the epicenter of the database with primary key as S_ID. It has several attributes regarding name, age, email and address. **STUDENT** entity has weak relationship with entities **GROUP**, **COURSE**, **CLASS** and **SCHEDULE**, as these entities are strong entities and don't use **STUDENT**'s primary key as their own primary key. Also **STUDENT** has strong relationship with **LIBRARY** and **RATING** as both are weak entities with **STUDENT**'s primary key (S_ID) as their own primary key.

Each **STUDENT** enrolls in a **COURSE** with respective attributes as fees, name etc. Each course is assumed to be comprised of respective **CLASS** (i.e programming classes like Python, Java etc) and **PROJECT** (with categories i.e Arduino, Rasp. Pi etc) as observed from the ER diagram. Strong relationship is observed as both **CLASS** and **PROJECT** use **COURSE**'s primary key (C_ID) as their own primary key.

GROUP entity assumes that each student belongs to a respective fixed group, which can have a maximum of 4 members. TABLE_ID is regarding the specific allocated table for each group, which in turn keeps the track of all **PARTS** assigned to the group. **PARTS** consists of info related to supplier and ORDER_NO, to keep a track of orders.

Each **INSTRUCTOR** can teach many **CLASS** (also a coding **CLASS** would have multiple **INSTRUCTOR**) and conduct many **PROJECT**, thus the relationship cardinality is reflected in diagram.

RATING is done by a respective student regarding **CLASS**, **PROJECT** and **INSTRUCTOR**, thus S_ID is used as the primary key.

INSTRUCTOR recommends multiple **TEXTBOOK** to respective CLASS_IDs. Also **LIBRARY** acts as a bridge entity between **STUDENT** and **TEXTBOOK**.

HOURS table keeps track of **INSTRUCTOR**'s project hours (HOURS_P) and class hours (HOURS_CLASS) that he/she taught with respect to DATE as primary key, in order to calculate TOTAL_WAGE.