**Weak Entities**

Entity types that do not have key attributes of their own are called weak entity types. In contrast, regular entity types that do have a key attribute are called strong entity types.

Entities belonging to a weak entity type are identified by being related to specific entities from another entity type in combination with one of their attribute values. This entity type is called the identifying or owner entity type.

Weak entity set means that entity set which is having no primary key. So unique identification of records or tuples are not possible.

Consider the entity type DEPENDENT, related to EMPLOYEE, which is used to keep track of the dependents of each employee via a 1:N relationship.

Suppose that a DEPENDENT entity is identified by the dependents first name and birthdate and the specific EMPLOYEE as its identifying entity type via the identifying relationship type DEPENDENT-OF.

Here weak entity type is: DEPENDENT

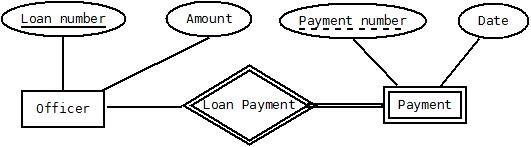
Identifying relationship is: DEPENDENTS-OF

In ER diagrams, both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines. The partial key attribute is underlined with a dashed or dotted line.

In case of weak entity sets, we have partial key.

Let us go through one example for better understanding.







Loan Number Amount Payment Number Payment date Payment amt

L1 100000 20 10/10 5000



L2 150000 20 10/10 5000

L3 200000 20 10/10 5000

We know that weak entity set is denoted by double rectangles. The name of the weak entity set is payment. Payment number is a partial key and it is dotted underline.

Entity Loan is having two attributes i.e. Loan number and amount. Loan number is primary key. Loan Payment is identifying relationship and is denoted by double diamonds.

Identifying relationship and weak entity set will be merged in the case of total participation. Single line arrow indicating that against one loan there will be multiple payments. We have taken a loan and we are going to pay instalments against the loan.

Let us consider I have three loans from bank say L1, L2 and L3. I have plan to pay 5000 as instalment against each loan. I have gone to bank and taken payment slip and containing number 20.

Here loan number is prime attribute. We have two entities Loan (Loan-number, amount) and Payment(Payment no, Payment date, Payment amount). Loan-Payment table is identifying relationship and weak entity set.

Loan-payment(loan-number, payment no, payment-date, payment-amount)

In the loan-payment table, loan-number and payment no will be the primary key and it become a strong entity.

Entries of this table as follows

L1 20 10/10 5000

L2 20 10/10 5000

L3 20 10/10 5000

In this way, weak entity set has been converted into strong entity set after total participation of identifying relationship.

**Relation Schema**

Relation schema defines the design and structure of the relation like it consists of the relation name, set of attributes/field names/column names. Every attribute would have values.

There is a student named Swathi, she is pursuing BCA, in the 2nd year, and belongs to Computer Science department (department no. 1) and has roll number 1601347 She is advised by Mrs. SS. If we want to represent this using database, we have to create a student table with name, gender, degree, year, department, department number, roll number and adviser as the attributes.

student (RollNo, name, degree, year, gender, deptNo, advisor)

This and other departments can be represented by the department table, having department ID, name, hod and phone as attributes.

department (deptId, name, hod, phone)

The course that a student has selected has a courseid, course name, credit and department number.

course (coursId, ename, credits, deptNo)

The professor would have an employee Id, name, gender, department no. and phone number.

professor (empId, name, gender, joining date, deptNo, phone)

We can have another table named enrollment, which has roll no, courseId, semester, year and grade as the attributes.

enrollment (rollNo, coursId, sem, year, grade)

Teaching can be another table, having employee id, course id, semester, year and classroom as attributes.

teaching (empId, coursed, sem, year, Classroom)

When we start courses, there are some courses which another course that needs to be completed before starting the current course, so this can be represented by the Prerequisite table having prerequisite course and course id attributes.

prerequisite (preReqCourse, courseId)