**4/16/2020**

Bad database designs:

1. Unnecessary redundancy: inefficient storage
2. Anomaly: conflicts in data, difficulties in maintenance.

Thus, EMP\_NO is a candidate key.

Assumption: Every department has only one manager.

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_NO** | **NAME** | **DEPT\_NO** | **MANAGER\_NO** |
| 101 | Lady Gaga | *D123* | *110* |
| 122 | Brad Pitts | *D123* | *110* |
| 140 | Lebron James | *D123* | *110* |
| 155 | Narendra Modi | D225 | 205 |
| 167 | Jennifer Lopez | D225 | 205 |
| 311 | John Smiths | D337 | 333 |

1. **Problem:**

**Update Anomaly:**

(a) 415 is the new manager of department D123

* Inefficiency in update.
* Potential inconsistency.

Assumption: Every department has only one manager.

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_NO** | **NAME** | **DEPT\_NO** | **MANAGER\_NO** |
| 101 | Lady Gaga | D123 | **415** |
| 122 | Brad Pitts | D123 | **415** |
| 140 | Lebron James | D123 | ***110 if not updated -> 415*** |
| 155 | Narendra Modi | D225 | 205 |
| 167 | Jennifer Lopez | D225 | 205 |
| 311 | John Smiths | D337 | 333 |

SQL: it works but not efficient.

UPDATE employee
SET MANAGER\_NO = 415
WHERE DEPT\_NO = 'D123';

Correct, not efficient

(b) Jennifer Lopez changes to working for department D337:

* Need to know the manager of D337.
* Potential inconsistency.

|  |  |  |  |
| --- | --- | --- | --- |
| **EMP\_NO** | **NAME** | **DEPT\_NO** | **MANAGER\_NO** |
| 101 | Lady Gaga | D123 | 110 |
| 122 | Brad Pitts | D123 | 110 |
| 140 | Lebron James | D123 | 110 |
| 155 | Narendra Modi | D225 | 205 |
| 167 | Jennifer Lopez | **D337** | *205 if not updated -> 333* |
| 311 | John Smiths | D337 | 333 |

The command:

UPDATE Employee
SET DEPT\_NO = 'D337'
WHERE NAME = 'Jennifer Lopez';

will produces inconsistent result.

You need to update both DEPT\_NO and MANAGER\_NO. However,

UPDATE Employee
SET DEPT\_NO = 'D337',
     MANAGER\_NO = (SELECT DISTINCT MANAGER\_NO FROM Employee WHERE DEPT\_NO = 'D337')
WHERE NAME = 'Jennifer Lopez';

will not work in MySQL as one cannot include a SELECT clause on the same table in the SET clause.

mysql> select max(actor\_id) from actor;

+---------------+

| max(actor\_id) |

+---------------+

| 200 |

+---------------+

1 row in set (0.07 sec)

mysql> INSERT INTO film\_actor

 -> VALUES (201,1,'2006-02-15 05:05:03');

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`sakila`.`film\_actor`, CONSTRAINT `fk\_film\_actor\_actor` FOREIGN KEY (`actor\_id`) REFERENCES `actor` (`actor\_id`) ON DELETE RESTRICT ON UPDATE CASCADE)

mysql>

Problem:

A student (stuId) may take a course (cid) and get a grade. A given has a unique credit.

UML/OO:

Class: student(stuId)

Class: course(cid, credit)

Student (many) to course (many) association: attribute: grade.

Relation model: attributes relations.

CId -> credit

(many) to (one)

CSCI5333.1 -> 3

CSCI4333.1 -> 3

X-> Y given an X, there is only one Y value.

CId -> credit
StuId, CId -> Grade is true

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| … | StuId | CId | Grade | phone | Course Credit |
|  | S1 | C1 | A | P1 | 3 |
|  | S1 | C1 | A (ok) | P2 | 3 |
|  | S1 | C1 | B (not allowed by StuId, CId -> Grade) |  | 3 (ok) |
|  |  | C1 |  |  | 4 (not allowed by CId -> credit) |