



TM

baaab**tra**

Mentoring Partner

Disclaimer: This presentation is prepared by trainees of baabtra as a part of mentoring program. This is not official document of baabtra –Mentoring Partner

Baabtra-Mentoring Partner is the mentoring division of baabte System Technologies Pvt . Ltd

Anomalies in Database



NAME



**ABHISHEK MENON121989CGMAIL
.COM**



**WWW.FACEBOOK.COM/
ABHISHEK H MENON**



**TWITTER.COM/ABHISHEK H
MENON**



**IN.LINKEDIN.COM/IN/ABHISHEK
H MENON**



Anomaly

Normalization is a technique for producing relational schema with the following properties.

- no information redundancy.
- no **anomalies**.

so we are clear when the process of reducing the **anomaly** comes into action.

What is an Anomaly?

- Definition
 - Problems that can occur in poorly planned, un-normalized databases where all the data is stored in one table (a flat-file database).

Types of Anomalies

- Insert
- Delete
- Update

Insert Anomaly

- An **Insert Anomaly** occurs when certain attributes cannot be inserted into the database without the presence of other attributes.

Insert Anomaly

Course_no	Tutor	Room	Room_size	En_limit
353	Smith	A532	45	40
351	Smith	C320	100	60
355	Clark	H940	400	300
456	Turner	H940	400	45

e.g. we have built a new room (e.g. B123) but it has not yet been timetabled for any courses or members of staff.

Delete Anomaly

- A **Delete Anomaly** exists when certain attributes are lost because of the deletion of other attributes.

Delete Anomaly

Course_no	Tutor	Room	Room_size	En_limit
353	Smith	A532	45	40
351	Smith	C320	100	60
355	Clark	H940	400	300
456	Turner	H940	400	45

e.g. if we remove the entity, course_no:351 from the above table, the details of room C320 get deleted. Which implies the corresponding course will also get deleted.

Update Anomaly

- An **Update Anomaly** exists when one or more instances of duplicated data is updated, but not all.

Update Anomaly

Course_no	Tutor	Room	Room_size	En_limit
353	Smith	A532	45	40
351	Smith	C320	100	60
355	Clark	H940	400	300
456	Turner	H940	400	45

e.g. Room H940 has been improved, it is now of RSize = 500. For updating a single entity, we have to update all other columns where room=H940.

A conceptual model

- Consider the following 'simple' conceptual data model:



Staff(Staff-id, Name, address, scale_point, Rate_of_pay, DOB..)

Student(Enroll-no, Name, Address, Olevelpoints..)

Course(Course-code, Name, Duration..)

The 'Translation' process

- Entities become relations.
- Attributes become attributes.
- Entity identifier (s) become primary key(s)
- Relationships are represented by additional foreign keys attributes in those relation that are at the 'M' end of the 1:M relationship.

The 'Staff' and 'Student' relations

- Staff(Staff-id, Name , Address, Rate-of-pay, DOB)

<u>Staff_id</u>	Name	Address	Scale point	Rate_of_pay	DOB

- Student(Enrol-no, Name, Address, Olevelpoints, Tutor);

<u>Enrol-no</u>	Name	Address	Olevelpoints	Tutor

NB: **Foreign Key** Tutor **references** Staff.Staff_id

The 'Staff' & 'Course' relations

- Staff

<u>Staff id</u>	Name	Address	Scalepoint	Rate_of_pay	DOB

- Course

<u>Course-code</u>	Name	Duration

NB: *according to the conceptual model specified, both the above entities have M:M relationships at the end.*

We must create an artificial link in between these two entities.

The 'Staff', 'Course' and 'Team' relations

- Staff

<u>Staff_id</u>	Name	Address	Scalepoint	Rate_of_pay	DOB

- Team

<u>Course_code</u>	<u>Staff_id</u>

- Course

<u>Course_code</u>	Name	Duration

The 'Staff', 'Course' and 'Team' relations

- In our new relation 'Team':
 - Primary key is the combination of Course_code and Staff_id.
 - Foreign key Course_code references Course.Course_code.
 - Foreign key Staff_id references Staff.Staff_id.

For 3 entities 4 relations

- Student

<u>Enrol_no</u>	Name	Address	Olevelpoints	Tutor

- Staff

<u>Staff_id</u>	Name	Address	Scalepoint	RateofPay	DOB

- Team

Course_code	Staff_id

- Course

Course_code	Name	Duration

But these tables are not free from the above anomalies!

Checking for anomalies

- every tuple unique?
- no hidden meaning from location?
- data cells atomic?
- For relations with single_attribute keys:
 - every attribute depends upon the primary key
- For relations with composite keys:
 - Every attribute depends upon the whole key(the components of this key are prime attributes?)

Relation fails

- Especially when we check dependency between various attributes in a relationship.
- So we must split that relation into two relations.
- And we should create a foreign key in the new table which references the primary key in the old table.

Problem arises

- The attribute '**Rate_of_pay**' does not depends on the '**Staff_id**' but on '**Scalepoint**' in the relation **Staff**.
- So the next step will be splitting the relation **Staff** into relation **Staff** and relation **Pay**.

Staff	<u>Staff_id</u>	Name	Address	Scalepoint	DOB
Pay	<u>Scalepoint</u>		<u>Rate of Pay</u>		

Where in new relation Pay;

Combination of **Scalepoint** and **rate_of_pay** is taken as primary key.

Foreign key **Scalepoint** references **Staff.Scalepoint**.

For 3 entities 5 relations

- Student

<u>Enrol_no</u>	Name	Address	Olevelpoints	Tutor

- Staff

<u>Staff_id</u>	Name	Address	Scalepoint	DOB

- Team

<u>Course_code</u>	Staff_id

- Course

<u>Course_code</u>	Name	Duration

- Pay

<u>Scalepoint</u>	<u>Rate of pay</u>

If this presentation helped you, please visit our
page facebook.com/baabtra and like it.

Thanks in advance.

www.baabtra.com | www.massbaab.com | www.baabte.com

Contact Us



+91 495 2295550



info@baabtra.com



+91 9539906339



www.baabtra.com



MP/1575A, NC Complex, Near Bus Stand
Mukkam-673602, Kozhikode Dt.