بسم الله الرحمن الرحيم

Database Programming

برمجة قواعد البيانات

عدد الساعات: ٢ نظري+٢ عملي

الرمز: ٣١٤ حسب

المتطلبات: ٢٢٣ حسب (مبادئ قواعد البيانات)

أستاذات/المادة: م. لندا عمر البدري - م. نجلاء حسن

المحاضرة الثالثة

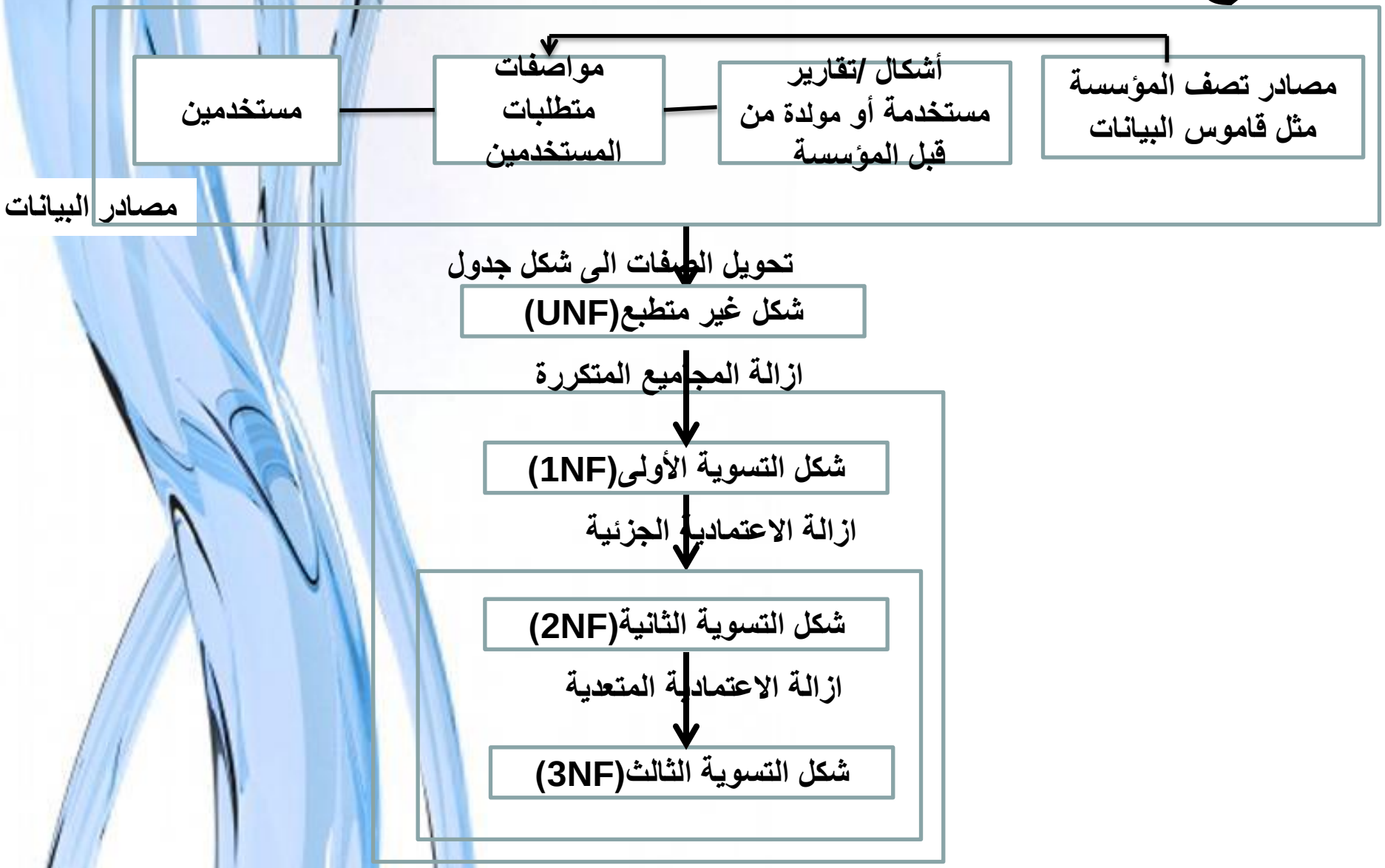
تابع التطبيع (التسوية)

Normalization

عملية التسوية NORMALIZATION PROCESS

- التسوية هي تقنية مستخدمة لتحليل العلاقات اعتماداً على مفتاحها الرئيسي primary key (أو المفاتيح المرشحة candidate keys) والاعتمادية الوظيفية.
- تحتوي التقنية على سلسلة من القواعد التي يمكن استخدامها لاختيار العلاقات بصورة مفردة حتى تكوين قاعدة بيانات تطبيعية Normalized
- عندما لا يتم تحقيق المطلب، فان العلاقة التي تنتهك المطلب ولا تحققه يجب تجزأتها الى علاقات تحقق بمفردها متطلبات التسوية.
- تنفذ التسوية على شكل سلسلة من الخطوات، تطابق كل خطوة صيغة تسوية معينه (1NF, 2NF, ...) لها صفات معروفة.

تابع: عملية التسوية NORMALIZATION PROCESS



صيغة التسوية الأولى First Normal From (1NF)

- صيغة العلاقة غير المتطبعة (UNF): عبارة عن جدول يحتوي على مجموعة أو أكثر من المكررات.
- صيغة التسوية الأولى: علاقة يكون فيها تقاطع كل صف وعمود يحتوي على قيمة واحدة فقط.
- لتحويل الجدول الى صيغة التسوية الأولى:

- الخطوة الاولى :نحدد ونزيل المجاميع المتكررة، وذلك بادخال بيانات ملائمة في الأعمدة الفارغة للصفوف التي تحتوي على بيانات مكررة(ملأ الفراغات بتكرار البيانات غير المكررة) يُشار الي هذا الاسلوب بتسطيح الجداول.
- الخطوة الثانية تحديد المفتاح الرئيسي . (وفي هذا المثال تم تحديد المفتاح الرئيسي (Emp-no, Proj-no

PROJ -NUM	PROJ-NAME	EMP-NUM	EMP-NAME	JOB-CLASS	CHG-HOUR	HOURS
15	Evergeen	103, 105 ,101, , 102 106	June Arbouhg	Elect.Engineer	$84.5	23.8
18	Amberwave	118 104 ,114, 112	Annellie	Application Designer	$48.10	24.6
22	Rolling	,104, 105	Alice k.Johnson	DB designer	$105.00	64.7

جدول في صيغة العلاقة الغير متطبعة

Un Normal Form UNF

PROJ-NUM	PROJ-NAME	EMP-NUM	EMP-NAME	JOB-CLASS	GHG-HOUR	HOURS
15	Evergeen	103	June Arbouhg	Elect.Engineer	$84.5	23.8
		101				
		105				
		106				
		102				
18	Amberwave	114	Annellie	Application Designer	$48.10	24.6
		118				
		104				
		112				
22	Rolling	105	Alice k.Johnson	DB designer	$105.00	64.7
		104				

PROJ-NUM	PROJ-NAME	EMP-NUM	EMP-NAME	JOB-CLASS	GHG-HOUR	HOURS
15	Evergeen	103	June Arbouhg	Elect.Engineer	$84.5	23.8
15	Evergeen	101	June News	D.BDesigner	$105.5	19.4
15	Evergeen	105	Alice k.Johnson	D.BDesigner	$105.5	35.7
15	Evergeen	106	William	Programmer	$35.75	12.6
15	Evergeen	102	David	System Analyst	$96.75	23.8
18	Amberwave	114	Annelise	Application Designer	$48.10	24.6
18	Amberwave	118	James	General support	$10.36	45.3
18	Amberwave	104	Anne Romors	System Analyst	$96.75	32.4
18	Amberwave	112	Darlene	DSS Analyst	$45.95	44.0
22	Rolling	105	Alice k.Johnson	DB designer	$105.00	64.7
22	Rolling	104	Anne Romors	System Analyst	$96.75	48.4

- الجدول السابق يكون في صيغة التسوية الأولى First Normal Form (1NF) .
- الجدول يكون في صيغة (1NF) عندما تكون جميع صفات المفتاح محددة ، وعندما تكون جميع الصفات الباقية معتمدة على المفتاح الرئيسي . ولا توجد مجاميع مكررة في الجدول (تقاطع كل صف وعمود يحتوي على قيمة واحدة فقط).
- الجدول في صيغة (1NF) يمكن أن يحتوي على اعتمادات وظيفية جزئية و متعدية .
- الجدول ذو المفتاح الرئيسي المكون من صفة مفردة لايمكن ان يحتوي على اعتمادية جزئية .

تابع خطوات التسوية

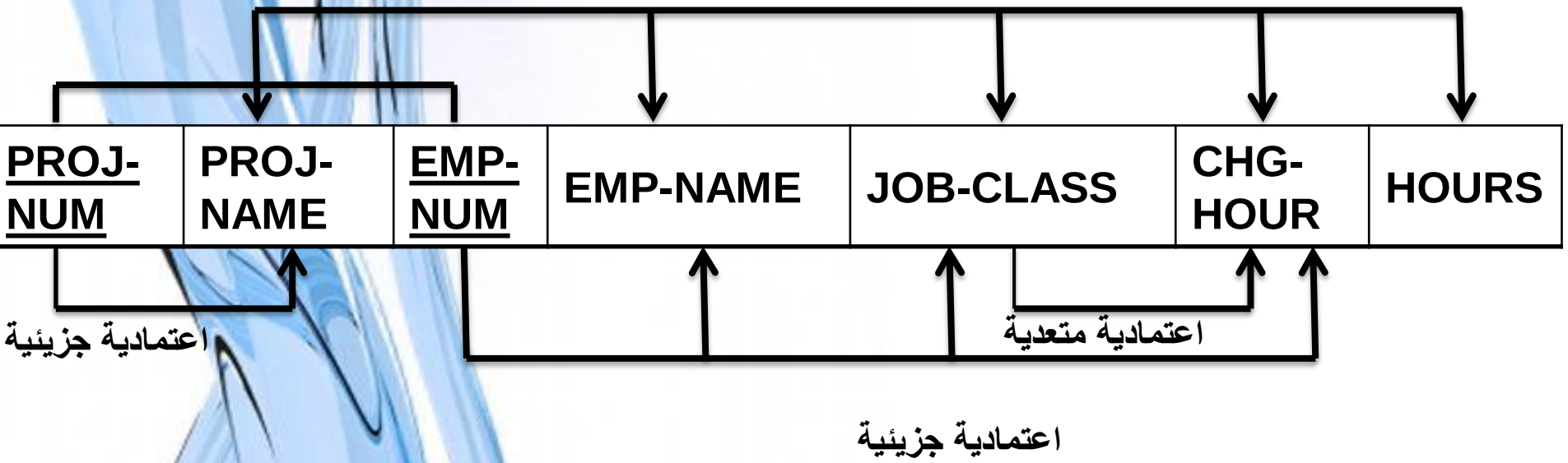
- الخطوة الثالثة : تحديد جميع المعتمدات :

PROJ_NUM,EMP_NUM ⟶ PROG_NAME, JOB_CLASS , CHG_HOUR,HOURS

PROG_NUM ⟶ PROG_NAME.

EMP_NUM ⟶ EMP_NAME,JOB_CLASS, CHG_HOUR

JOB_CLASS ⟶ CHG_HOUR



مخطط الاعتمادية صيغة التسوية الاولي

صيغة التسوية الثانية Second Normal Form (2NF)

- خطوات التحويل من صيغة التسوية 1NF الى 2NF :

١- تحديد جميع مكونات المفتاح:

PROJ_NUM

EMP_NUM

PROJ_NUM , EMP_NUM

كل جزء من المفتاح يصبح في جدول جديد هي:

PROJECT, EMPLOYEE,ASSIGN

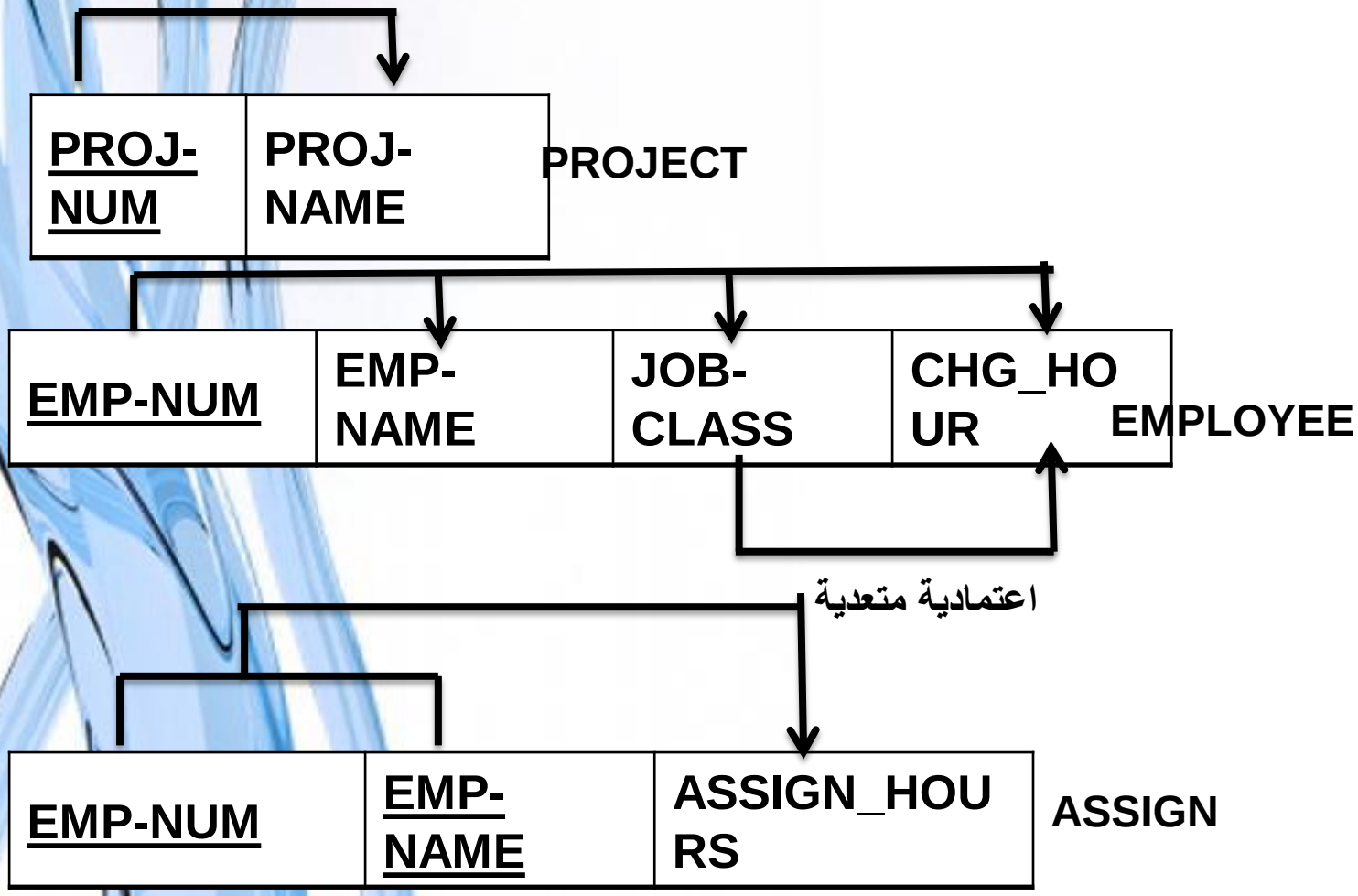
تابع صيغة التسوية الثانية Second Normal Form (2NF)

٢- تحديد الصفات المعتمدة

PROJECT(PROJ_NUM, PROJ_NAME)

EMPLOYEE(EMP_NUM,EMP_NAME, JOB_CLASS,CHG_HOUR)

ASSIGN(PROJ_NUM,EMP_NUM, ASSIGN_HOURS)



صيغة التسوية الثانية

- يكون الجدول في صيغة التسوية الثانية اذا كان :

١. في صيغة التسوية الأولى .

٢. لم يتضمن اعتمادية جزئية .

- يمكن للجدول في صيغة التسوية الثانية ان يحتوي على اعتمادية متعدية بحيث تكون واحدة أو أكثر من الصفات معتمدة وظيفياً على صفات غير مفتاحية .
- يترتب عن وجود اعتمادية متعدية مشاكل الاضافة والتعديل في البيانات (مثال اذا تغيرت الكلفة لكل ساعة لتصنيف العمل الذي يشغل من قبل عدد من الموظفين، فيجب عمل ذلك التغيير لكل واحد من الموظفين)

صيغة التسوية الثالثة Third Normal Form (3NF)

- يمكن القضاء على مشاكل البيانات المتكونة في صيغة (2NF) من خلال :

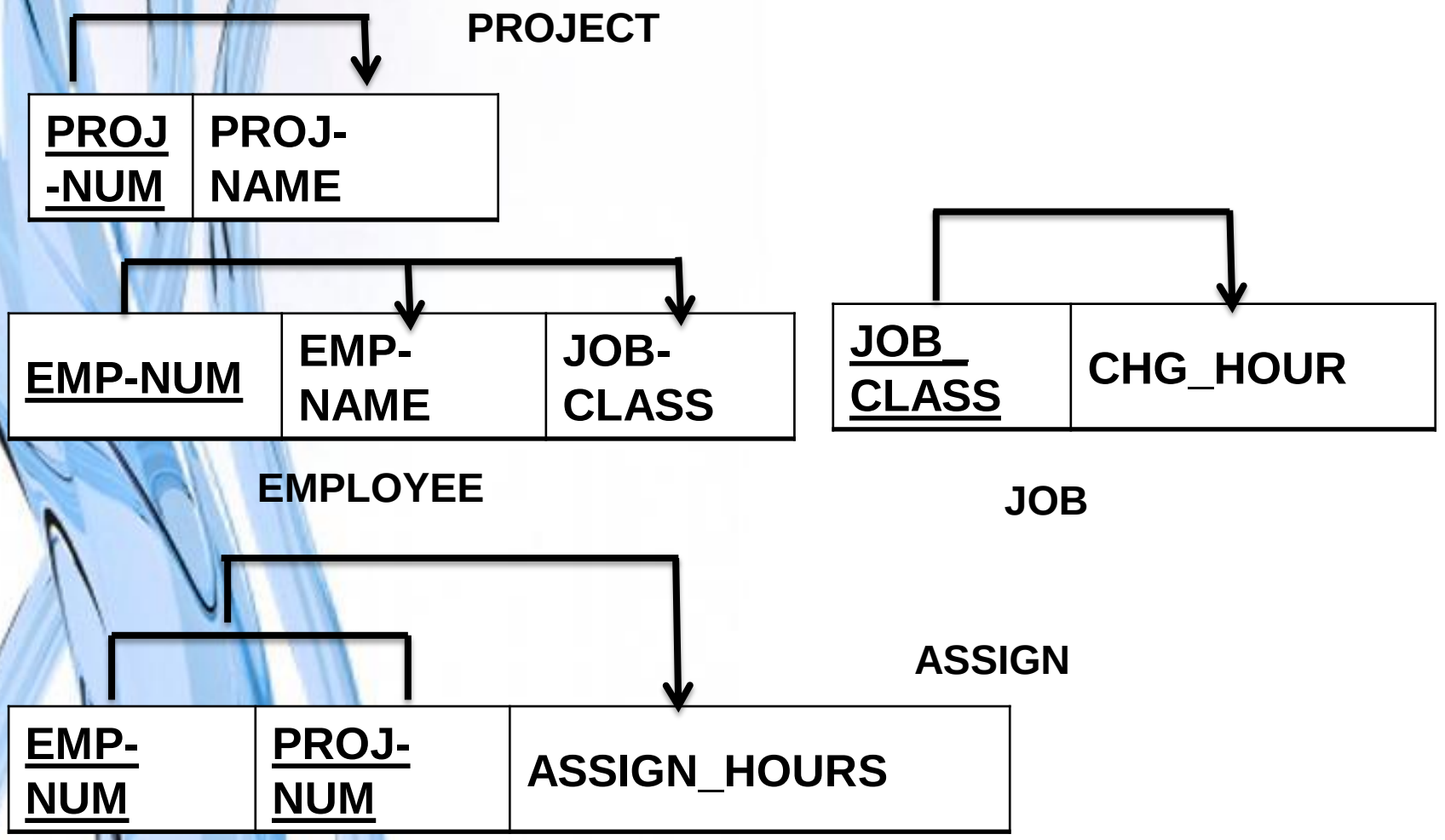
١. تحديد كل مقرر جديد ، لكل اعتمادية متعدية نكتب مقررها كمفتاح رئيسي لجدول جديد، مثال : (JOB_CLASS) .

٢. تحديد الصفات التابعة ، حدد الصفات التابعة لكل مقرر محدد في الخطوة ١ ، وحدد التبعية

مثال CHG-HOUR ⟶ JOB_CLASS .

تسمية الجدول باسم يعكس محتوياته ووظيفته مثال (JOB)

٣. ازالة الصفات التابعة من الاعتمادية المتعدية من كل جدول يحتوي على مثل هذه الاعتمادية المتعدية ، في المثال نتخلص من ال CHG-HOUR.



صيغة التسوية الثالثة

- بعد أن يتم التحويل تحتوي قاعدة البيانات على اربعة جداول :

PROJECT(PROJ_NUM, PROJ_NAME)

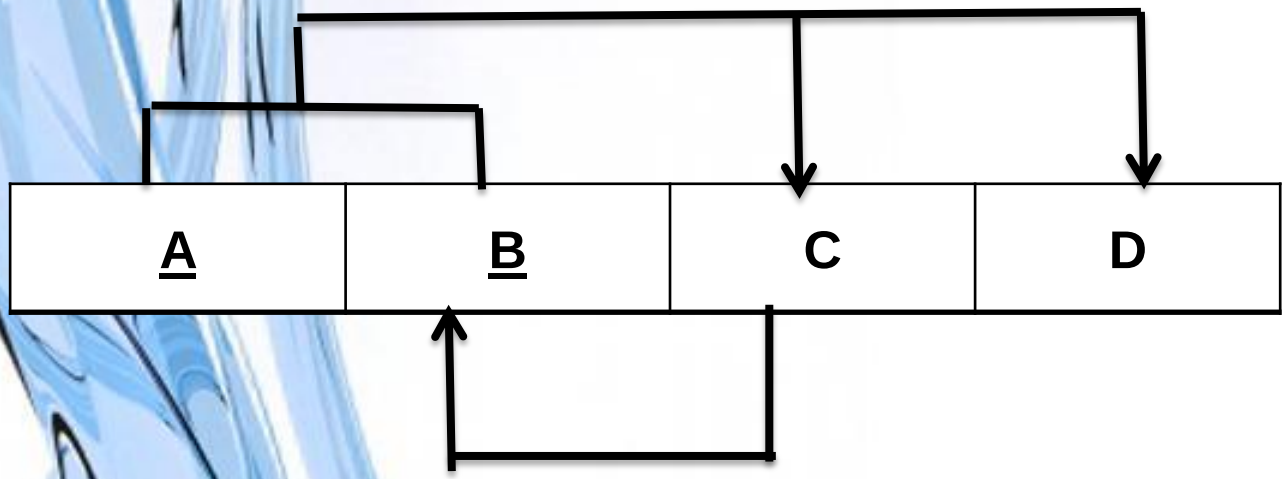
ASSIGN(PROJ_NUM,EMP_NUM,ASSIGN_HOURS)

EMPLOYEE(EMP_NUM,EMP_NAME, JOB_CLASS)

JOB(JOB-CLASS,CHG_HOUR)

صيغة تسوية بويس -كود (BCNF)

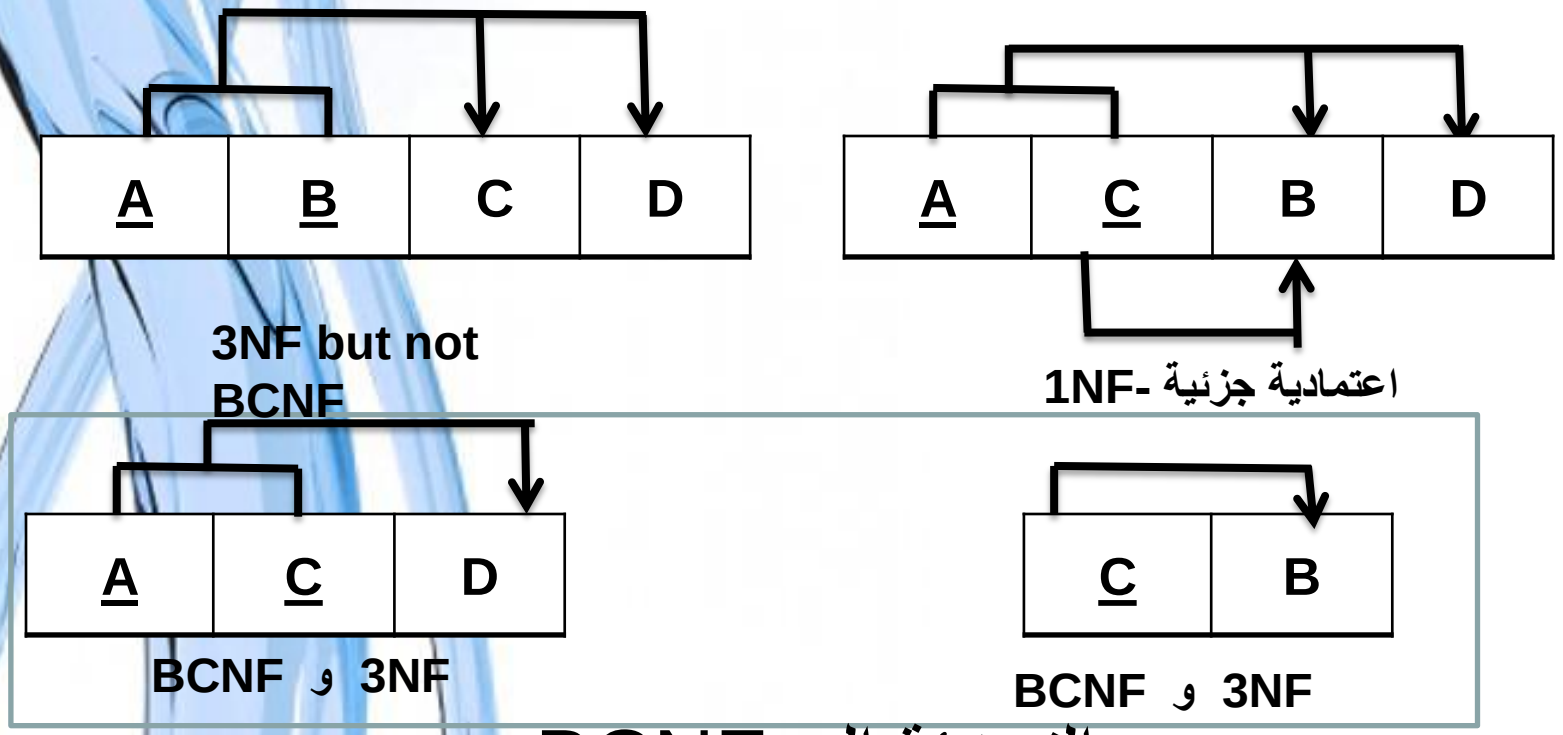
- صيغة BCNF حالة خاصة من صيغة 3NF .
- يكون الجدول في صيغة BCNF اذا كان المقرر في الجدول هو مفتاح مرشح
- تكون صيغ 3NF و BCNF متكافئة اذا احتوى الجدول على مفتاح مرشح واحد .
- يمكن لجدول أن يكون في حالة 3NF وليس في صيغة BCNF وذلك عند وجود صفة غير مفتاحية هي المقرر لصفة مفتاحية ، الشكل التالي يوضح ذلك .



جدول في حالة 3NF وليس في حالة BCNF

.

- لتحويل الجدول السابق الى جداول صيغة 3NF وصيغة BCNF نتبع الخطوات:
- تغيير المفتاح الرئيسي الى A+C ، لأن الاعتمادية C→B تعني أن C مجموعة رئيسية الى B. (وبذلك يكون الجدول في صيغة 1NF)
- نتبع طرق التجزئة القياسية لاعطاء النتائج التالية:



التجزئة الى BCNF

صيغ التسوية الأعلى

- يكون الجدول في صيغة التسوية الرابعة 4NF اذا كان في صيغة التسوية الثالثة 3NF ، وليس له مجاميع متعددة من التبعات المتعددة القيم ، (ان يكون أحد العاملين له مهام متعددة ، وايضاً متواجد في مؤسسات خدمة متعددة) .
- مناقشة صيغة ال 4NF هي نظرية أكثر مما تكون عملية (ينطبق هذا على صيغ التسوية الأعلى مثل 5NF)، خاصةً اذا تم التأكد بان الجداول تطبق القاعدتين التاليتن :

١. جميع الصفات معتمدة على المفتاح الرئيسي ولكن مستقلة الواحدة عن الأخرى .
٢. لا يحتوى الصف على اثنان او أكثر من الحقائق المتعددة القيم عن أي كينونة .

عكس التسوية(فتح التسوية)
De Normalization

- لان الجداول يتم تجزئتها من أجل تحقيق متطلبات التسوية فأن عدد جداول قاعدة البيانات يأخذ في الإزدياد ويكون مرتبط بهذه الزيادة عمليات الادخال والاخراج للقرص الاضافية ومنطق المعالجة ، والتي تؤدي جميعها الي تقليص سرعة النظام وبالتالي قد تكون هناك ظروف عرضية تسمح ببعض الدرجات في عكس التسوية من اجل زيادة سرعة المعالجة .
- نقاء التسوية غالباً من الصعوبة تحمله في بيئة قاعدة البيانات الحديثة ، ان التقاطع بين كفاءة التصميم ومتطلبات المعلومات وسرعة المعالجة يمكن حله من خلال عكس التسوية .

PL/SQL تطبيق

Declaring variables and constants

Variables and Data Types

Variables

– Used to store numbers, character strings, dates, and other data values

– Avoid using keywords, table names and column names as variable names

– Must be declared with data type before use:

variable_name data_type_declaration;

Scalar Data Types

- Represent a single value

Data Type	Description	Sample Declaration
VARCHAR2	Variable-length character string	`current_s_last VARCHAR2(30);`
CHAR	Fixed-length character string	`student_gender CHAR(1);`
DATE	Date and time	`todays_date DATE;`
INTERVAL	Time interval	`curr_time_enrolled INTERVAL YEAR TO MONTH;` `curr_elapsed_time INTERVAL DAY TO SECOND;`
NUMBER	Floating-point, fixed-point, or integer number	`current_price NUMBER(5,2);`

Table 4-2 Scalar database data types

Scalar Data Types

Data Type	Description	Sample Declaration
Integer number subtypes (BINARY_INTEGER, INTEGER, INT, SMALLINT)	Integer	`counter BINARY_INTEGER;`
Decimal number subtypes (DEC, DECIMAL, DOUBLE PRECISION, NUMERIC, REAL)	Numeric value with varying precision and scale	`student_gpa REAL;`
BOOLEAN	True/False value	`order_flag BOOLEAN;`

Table 4-3 General scalar data types

Scalar Data Types

- Variables are declared in PL/SQL using the syntax

<variable-name> <datatype> [not null] [:=<initial-value>]

- Constants are declared as follows:

<constant-name> constant <datatype> := <value>;

Examples

```
a   binary_integer;
cno  number(5) not null :=1111;
cname   varchar2(30);
commission  real(5,2) :=12.5;
maxcolumn constant integer(2) :=30;
Hired_date   date;
Done   boolean;
```

Only one variable can be declared at a time!

Anchored Data Types

- Anchored data types are determined by looking up another object's data type, which could be a column in the database.
- The anchored declarations have the syntax:

<variable-name> <object>%type [not null] [:=<initial-value>]

- where <object> is another previously declared PL/SQL variable or a database column.

Anchored Data Types (cont.)

Examples

Cnum customers.cno%type;

Cname customers.cname%type;

commission real(5,2) :=12.5;

X commission%type;

Composite Data Types

- PL/SQL provides two composite data types: PL/SQL tables and records.
- PL/SQL records can be table based, cursor based, or programmer defined.
- . Their declaration syntaxes are:

❑ <record-var-name> <table-name>%rowtype;

❑ <record-var-name> <cursor-name>%rowtype;

❑ type <type-name> is record

(<field1> <datatype1>,

<field2> <datatype2>, … ,

<fieldN> <datatypeN>);

Composite Data Types (cont.)

- PL/SQL tables are similar to database tables, except that they always consists of just one column indexed by binary integers. These tables have no bound and grow dynamically much like database tables.

Executing a PL/SQL Program in SQL*Plus

```
--PL/SQL program to display the current date
DECLARE
    todays_date DATE;
BEGIN
    todays_date := SYSDATE;
    DBMS_OUTPUT.PUT_LINE('Today''s date is ');
    DBMS_OUTPUT.PUT_LINE(todays_date);
END;
```

Figure 4-2 PL/SQL program commands

SELECT Statements

```
DECLARE
   v_sname   VARCHAR2(10);
   v_rating NUMBER(3);
BEGIN
   SELECT   sname, rating
    INTO    v_sname, v_rating
  FROM Salary
WHEREsid = 112;
 dbms_output.put_line('v_sname='||v_sname);
 dbms_output.put_line('v_rating='||v_rating);
END;
/
```

Sid	sname	rating
112	Hala	500
113	Mona	600

- INTO clause is required.
- **Query must return exactly one row.**
- **Otherwise, a NO_DATA_FOUND or TOO_MANY_ROWS exception is thrown**

THE END