



Database Management

BU.330.770

Session 3

Instructor: Changmi Jung, Ph.D.



Announcement

» We have Quiz #2 next week

- Canvas > Quizzes or Modules > Week 4
- Includes class contents from week 2 and week 3

» The logistics are the same

- Check your network, Respondus Lockdown Browser
- After submission, you may leave the classroom and come back to join the class within the given amount of time

» More questions than Quiz #1 + short answer questions about SQL coding



Constraints



Session Objectives (1/2)

- » Understand the purpose of constraints in a table
- » Distinguish among **PRIMARY KEY**, **FOREIGN KEY**, **UNIQUE**, **CHECK**, and **NOT NULL** constraints and the appropriate use for each constraint
- » Understand how constraints can be created when creating a table or modifying an existing table
- » Distinguish between creating constraints at the column level and table level



Session Objectives (2/2)

- » Create **PRIMARY KEY** constraints for a single column or a composite primary key
- » Create a **FOREIGN KEY** constraint
- » Create a **UNIQUE** constraint
- » Create a **CHECK** constraint
- » Create a **NOT NULL** constraint using the ALTER TABLE...MODIFY command
- » Include constraints during table creation
- » Use **DISABLE** and **ENABLE** commands
- » Use the **DROP** command



Constraints

- » Rules used to enforce business rules, practices, and policies
- » Rules used to ensure accuracy and integrity of data

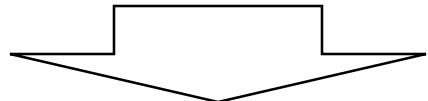
Order date	Nov. 6, 2024
Shipping date	Nov 2, 2024

?



Book's ISBN	(Null)
-------------	--------

?



Constraints play the role that prevents those erroneous data from being entered.



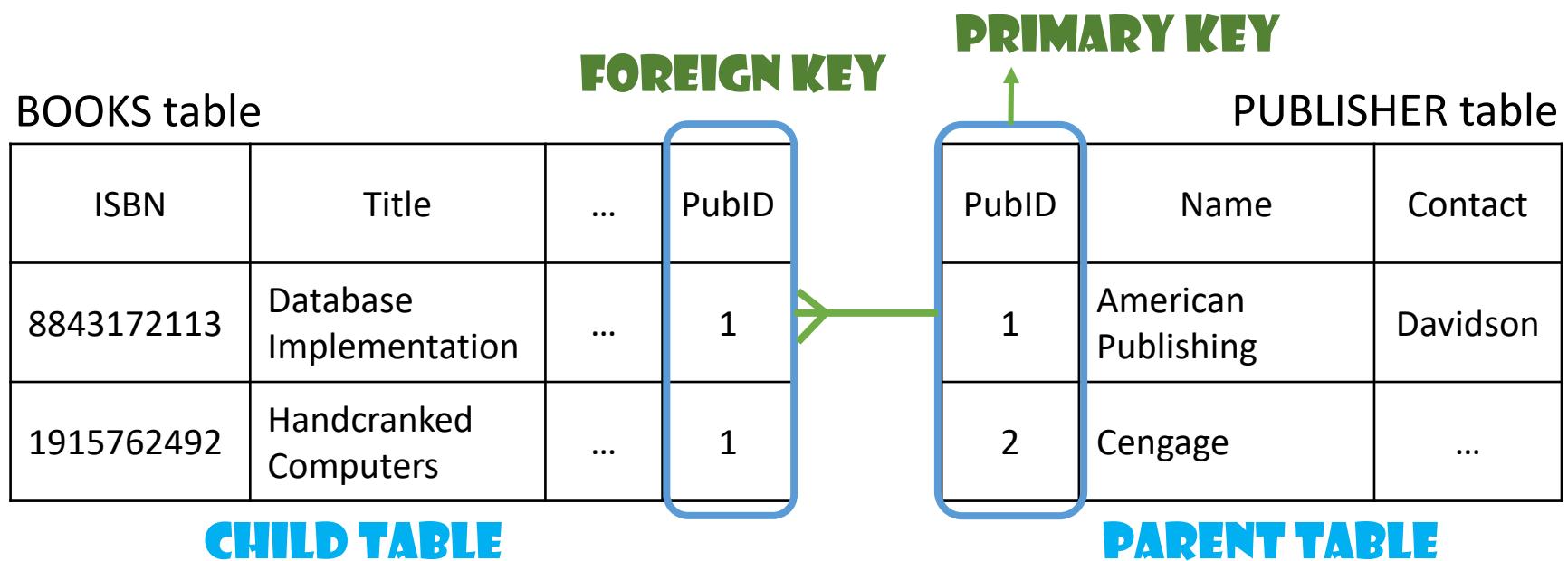
Constraint Types

Constraint	Description
PRIMARY KEY	Determines which column(s) uniquely identifies each record. The primary key Can't be Null , and the data values must be Unique .
FOREIGN KEY	In a one-to-many or parent-child relationship, the constraint is added to the "child (many)" table . The constraint ensures that if a value is entered in a specified column, it must already exist in the "one" table (parent table), or the record isn't added.
UNIQUE	Ensures that all data values stored in a specified column are unique. The UNIQUE constraint differs from the PRIMARY KEY constraint in that it allows NULL values .
CHECK	Ensures that a specified condition is true before the data value is added to a table. For example, an order's ship date can't be earlier than its order date.
NOT NULL	Ensures that a specified column can't contain a NULL value. The NOT NULL constraint can be created only with the column-level approach to table creation.



Recap: Relating Tables

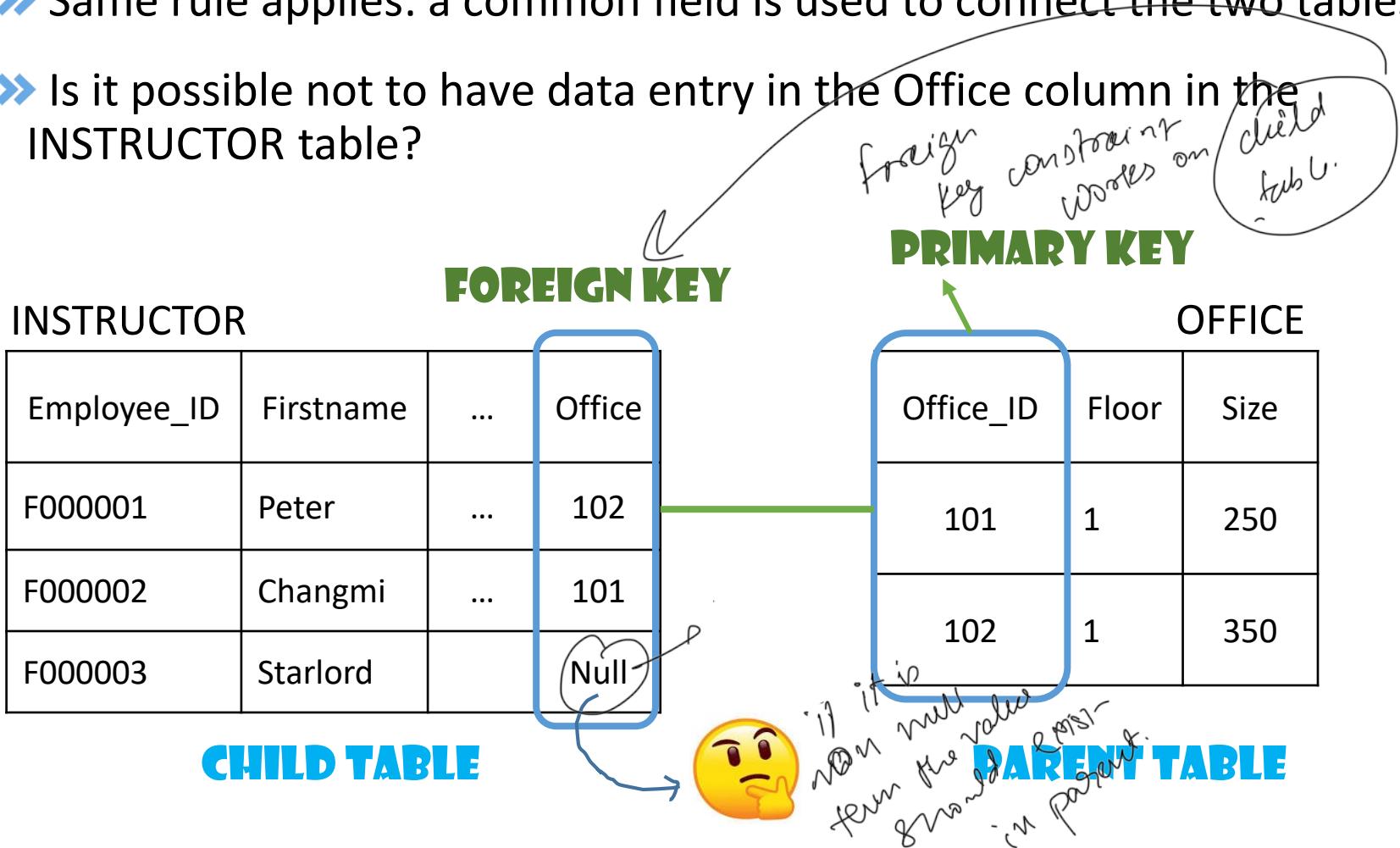
- » Tables are linked through a common field
- » A common field is usually a primary key in one table and a foreign key in the other table (many-side)
- » One side is a '**Parent table**', and many-side is a '**Child table**'





What about One-to-One?

- » Same rule applies: a common field is used to connect the two tables
- » Is it possible not to have data entry in the Office column in the INSTRUCTOR table?





Creating Constraints - Naming

- » Use the optional CONSTRAINT keyword during creation to assign a name → good practice!
 - Conventional naming is 'tablename_columnname_constrainttype'
 - Example: books_isbn_pk (primary key constraint applied to the isbn column from the books table)
- » You may let the server name the constraint using the default format SYS_Cn (but not recommended)
- » Informative names can assist in debugging



Creating Constraints - Approach

» Create Constraints During Table Creation

- Column-level approach
- Table-level approach

» Add Constraints to an Existing Table

- Modifying the existing table to add constraints



Creating Constraints at the Column Level (in Create Table)

- » If a constraint is being created at the column level, the constraint applies to the column specified

*columnname [CONSTRAINT *constraintname*] constrainttype,*

Example:

```
CREATE TABLE test1
(column1 NUMBER(3) CONSTRAINT test1_column1_pk PRIMARY KEY,
column2 VARCHAR2(12) NOT NULL,
column3 NUMBER(5,2));
```



We don't need to give constraint names to NOT NULL constraint



Creating Constraints at the Table Level (in Create Table)

- » Approach can be used to create any constraint type except for **NOT NULL**
- » Required if a constraint is associated with **multiple columns**

[CONSTRAINT *constraintname*] *constrainttype* (*columnname*, ...),

```
CREATE TABLE test2
(column1 NUMBER(3),
column2 VARCHAR2(12) NOT NULL,
column3 NUMBER(5),
column4 VARCHAR2(6),
CONSTRAINT test2_column1_pk PRIMARY KEY (column1),
CONSTRAINT test2_column3_4_uk UNIQUE (column3, column4)
);
```

term suffix helps understand the constraint type.

Example





Creating Constraints at the Table Level: Composite Primary Key

» What if we have a composite primary key?

```
CREATE TABLE test3
(column1 NUMBER(3),
column2 NUMBER(4),
column3 VARCHAR2(12) NOT NULL,
column4 VARCHAR2(6),
CONSTRAINT test3_column1_2_pk PRIMARY KEY (column1, column2),
CONSTRAINT test3_column4_uk UNIQUE (column4)
);
```

Example



Enforcement of Constraints

- » All constraints are enforced on a row level
- » If a data value violates a constraint, the entire row is rejected

PK Constraint		Unique Constraint		
Hero_ID	H_name	Zip	FirstName	LastName
001	Batman	20212	Bruce	Wayne
002	Ironman	90012	Tony	Stark
003	Superman	90012	Clark	Kent



which one violates constraint(s)?

Then what happens?

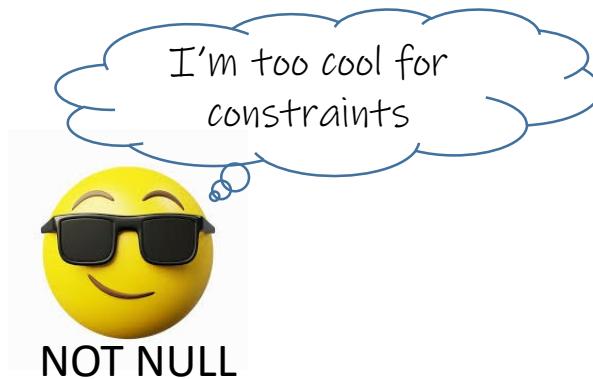


Adding Constraints to Existing Tables

» Constraints are added to an existing table with the ALTER TABLE command

» All constraints are added using ADD clause except for
NOT NULL

» Add a NOT NULL constraint using MODIFY clause





Adding the PRIMARY KEY Constraint

- » Ensures that columns do not contain duplicate or NULL values
- » Only one primary key constraint per table is allowed

ALTER TABLE *tablename*

ADD [CONSTRAINT *constraintname*] PRIMARY KEY (*columnname*);

*to add a
new column
or constraint*

*and UN
MODIFY
to change
existing column
definition.*



Let's Create a Test Table First

Worksheet Query Builder

```
CREATE TABLE customers1
AS (SELECT *
   FROM customers);
```

Script Output X

Task completed in 0.666 seconds

Table CUSTOMERS1 created.

» When creating a table from a subquery, most constraints do not copy into the new table, customers1. Only Not Null constraints are copied.

Compare the Constraints: customers vs. customers1



- » Identify constraints in each table by using the data dictionary, `USER_CONSTRAINTS`

Worksheet Query Builder

```
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'CUSTOMERS';
```

Script Output Query Result

SQL | All Rows Fetched: 4 in 0.052 seconds

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 SYS_C0022028	C	"LASTNAME" IS NOT NULL
2 SYS_C0022029	C	"FIRSTNAME" IS NOT NULL
3 CUSTOMERS_REGION_CK	C	region IN ('N', 'NW', 'NE', 'S', 'SE', 'SW', 'W', 'E')
4 CUSTOMERS_CUSTOMER#_PK	P	(null)

Refer to page 45

Only 'NOT NULL's in the new table, CUSTOMERS1



ADD CONSTRAINT Command

Worksheet Query Builder

```
ALTER TABLE customers1
ADD CONSTRAINT customers1_customer#_pk PRIMARY KEY (customer#);
```

constraint name constraint type column name

Query Result Script Output Task completed in 0.113 seconds

```
Table CUSTOMERS1 altered.
```

Now, let's check the constraints in CUSTOMERS1 by using
'user_constraints'



Constraint Checked with Data Input

Worksheet Query Builder

```
INSERT INTO customers1 (customer#, lastname, firstname, region)
VALUES (1020, 'PADDY', 'JACK', 'NE');
```

Query Result x Script Output x

Task completed in 0.119 seconds

```
Error starting at line : 54 in command -
INSERT INTO customers1 (customer#, lastname, firstname, region)
VALUES (1020, 'PADDY', 'JACK', 'NE')
Error report -
ORA-00001: unique constraint (ADMIN.CUSTOMERS1_CUSTOMER#_PK) violated
```

Constraint name referred in error message

Adding PRIMARY KEY Constraint for Composite Key



- » List column names within parentheses separated by commas

Worksheet Query Builder

```
CREATE TABLE orderitems1
AS (SELECT *
   FROM orderitems);

ALTER TABLE orderitems1
ADD CONSTRAINT orderitems1_order#_item#_pk PRIMARY KEY (order#, item#);
```

Query Result Script Output Task completed in 0.111 seconds

composite primary key

Table ORDERITEMS1 created.

Table ORDERITEMS1 altered.



Adding the FOREIGN KEY Constraint

- » Requires a value to exist in the referenced column of another table
- » Enforces **referential integrity**
- » NULL values are allowed
- » Maps to the PRIMARY KEY in the parent table

A book's publisher does not exist in the Publisher table?



```
ALTER TABLE tablename
```

```
ADD [CONSTRAINT constraintname] FOREIGN KEY (columnname)  
REFERENCES referenced_tablename (referenced_columnname);
```

Must be a primary key of the referenced table



Referential Integrity

A new order is placed, and thus a new record is being inserted into ORDERS.

FOREIGN KEY		ORDERS	
Order#	Customer#	Orderdate
1021	2001	05-Feb-24	...

Reference its parent table CUSTOMERS to check if customer# 2001 exists

PRIMARY KEY			
CUSTOMERS			
Customer#	Lastname	Firstname
2001	Jung	Changmi	...

If it doesn't exist in CUSTOMERS, the new order is rejected.



FOREIGN KEY Constraint Example

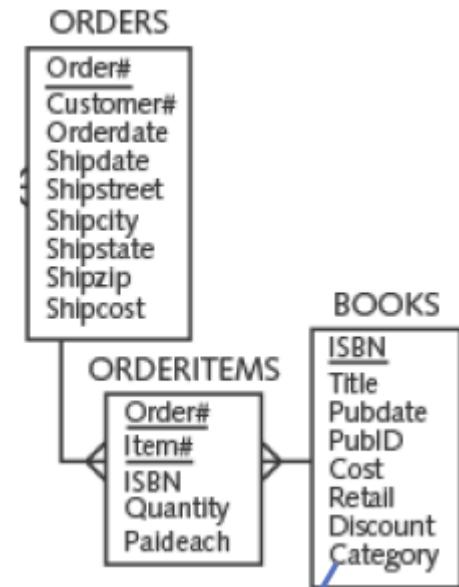
Worksheet Query Builder

```
ALTER TABLE orderitems1
ADD (CONSTRAINT orderitems1_order#_fk FOREIGN KEY (order#)
      REFERENCES orders (order#),
      CONSTRAINT orderitems1_isbn_fk FOREIGN KEY (isbn)
      REFERENCES books (isbn));
```

Query Result Script Output

Task completed in 0.097 seconds

Table ORDERITEMS1 altered.



Adding two constraints



Test FOREIGN KEY Constraint

Worksheet Query Builder

```
INSERT INTO orderitems1
VALUES (1030, 1, 1059831198, 2, 30.00);
```

This order# doesn't exist in its parent table, 'ORDERS'

Query Result Script Output

| Task completed in 0.111 seconds

```
Error starting at line : 81 in command -
INSERT INTO orderitems1
VALUES (1030, 1, 1059831198, 2, 30.00)
Error report -
ORA-02291: integrity constraint (ADMIN.ORDERITEMS1_ORDER#_FK) violated
```

parent key not found



Deletion of Parent Record

- » You cannot delete a value in a parent table referenced by a row in a child table (ex., order# in ORDERS is referenced by order# in ORDERITEMS)
- » Use **ON DELETE CASCADE** keywords when creating FOREIGN KEY constraint – it automatically deletes any corresponding records of the child table when a row in a parent table is deleted

```
ALTER TABLE bookauthor
```

```
ADD CONSTRAINT bookauthor_authorid#_fk FOREIGN KEY (authorid#)
```

```
    REFERENCES author (authorid) ON DELETE CASCADE;
```

- » Not recommended though



WE
WITH
Caution



Deletion of Parent Table

- » You cannot delete (drop) a parent table referenced by a child table
- » Drop the child table and then drop the parent table, usually.
- » Or, use **CASCADE CONSTRAINTS** option in DROP TABLE statement – it deletes the Foreign Key constraint in the child table and then deletes the parent table

```
DROP TABLE tablename  
  CASCADE CONSTRAINTS;
```



Using the UNIQUE Constraint

- » No duplicates are allowed in the referenced column
- » NULL values are permitted – performs a check on the data only if a value is entered into the column

The screenshot shows a database management interface with two tabs at the top: "Worksheet" and "Query Builder". The "Worksheet" tab is selected. In the main area, there is a code editor containing the following SQL command:

```
ALTER TABLE books
ADD CONSTRAINT books_title_uk UNIQUE (title);
```

Below the code editor is a toolbar with icons for "Query Result" and "Script Output". The "Query Result" tab is active, showing the message "Task completed in 0.123 seconds".

Table BOOKS altered.



Test UNIQUE Constraint

Worksheet Query Builder

```
INSERT INTO books (isbn, title)
VALUES (1212121212, 'SHORTEST POEMS');
```

Existing title in BOOKS table

Query Result Script Output

Task completed in 0.087 seconds

```
Error starting at line : 138 in command -
INSERT INTO books (isbn, title)
VALUES (1212121212, 'SHORTEST POEMS')
Error report -
ORA-00001: unique constraint (ADMIN.BOOKS ISBN PK) violated
```



Using the CHECK Constraint

- » Ensures updates and additions must meet specified condition

```
ALTER TABLE tablename
ADD [CONSTRAINT constraintname] CHECK (condition);
```

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a SQL script is being run to add a CHECK constraint to the 'orders' table. The constraint ensures that the 'shipdate' is greater than or equal to the 'orderdate'. The 'Script Output' tab at the bottom shows the message 'Table ORDERS altered.' indicating the success of the operation.

The condition cannot reference certain built-in functions, such as SYSDATE, or refer to values stored in other rows. But it can compare values in the same row.

Order date	Nov. 8, 2024
Shipping date	Nov 6, 2024

Ensures this doesn't happen



CHECK Constraint Conditions

- » Less than (<): e.g. **retail < 300**
- » Greater than (>): e.g. **cost > 0**
- » Range (BETWEEN): e.g. **retail BETWEEN 0 and 300**
- » List of values (IN): e.g. **region IN ('NE', 'SE', 'NW', 'SW')**

Worksheet Query Builder

```
ALTER TABLE acctmanager
ADD CONSTRAINT acctmanager_region_ck CHECK (region IN ('NE', 'SE', 'NW', 'SW'));

SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'ACCTMANAGER';
```

Script Output Query Result

SQL | All Rows Fetched: 3 in 1.76 seconds

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 ACCTMANAGER REGION CK	C	region IN ('NE', 'SE', 'NW', 'SW')
2 ACCTMANAGER AMID PK	P	(null)
3 ACCTMANAGER AMEDATE UK	U	(null)



Using the NOT NULL Constraint

- » The NOT NULL constraint is a special CHECK constraint with IS NOT NULL condition
- » Can only create it at the column level during table creation
- » Can only add it by using ALTER TABLE...**MODIFY** command to an existing table
- » Included in the DESCRIBE command output

```
ALTER TABLE tablename
```

```
    MODIFY (columnname [CONSTRAINT constraintname] NOT NULL);
```



NOT NULL Constraint Example

Worksheet Query Builder

```
ALTER TABLE orders
MODIFY customer# CONSTRAINT orders_customer#_nn NOT NULL;
```

Script Output Query Result Task completed in 0.108 seconds

Table ORDERS altered.

Worksheet Query Builder

```
ALTER TABLE acctmanager
MODIFY (amfirst NOT NULL,
        amlast NOT NULL);
```

Query Result Script Output Task completed in 0.04 seconds

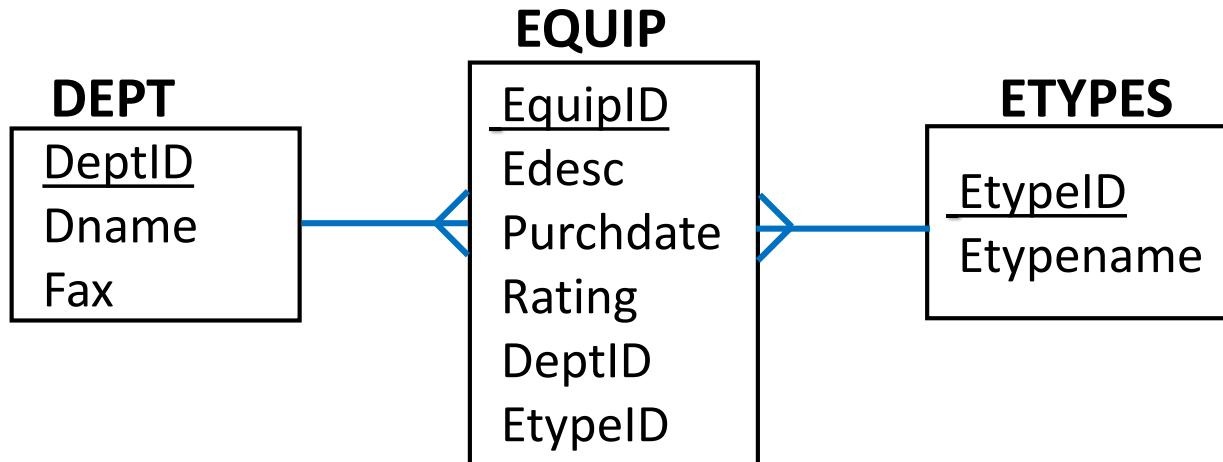
Check with user_constraints

Table ACCTMANAGER altered.



Let's Practice

JustLee Books would like to create some new tables to store office equipment inventory data (printers, monitors, etc.).

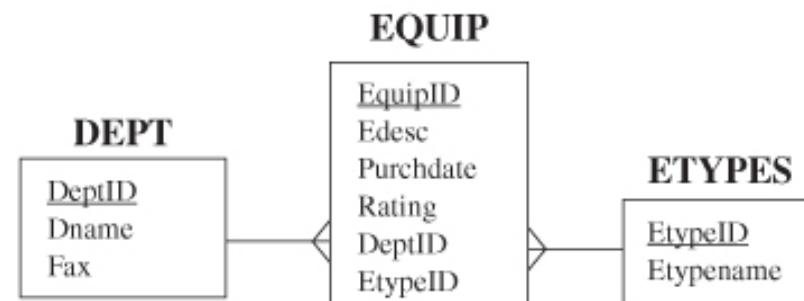


After analyzing the data requirements, the following list of requirements was developed (next slide). Those will be addressed with constraints.



The Requirements

1. Each department name must be unique
2. Each department must be assigned a name
3. Each equipment type name must be unique
4. Each equipment type must be assigned a name
5. Each equipment item must be assigned a valid department
6. If an equipment item is assigned a type, it must be a valid type
7. Valid rating values for equipment are A, B, and C



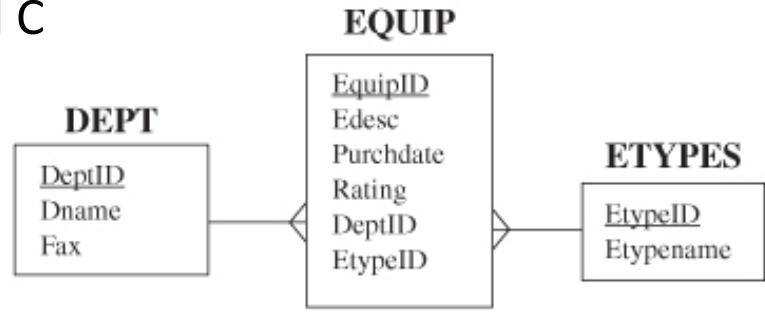


The Requirements – Which Constraints?

1. Each department name must be unique
2. Each department must be assigned a name *Non null*
3. Each equipment type name must be unique
4. Each equipment type must be assigned a name *Non null*
5. Each equipment item must be assigned a valid department

6. If an equipment item is assigned a type, it must be a valid type

7. Valid rating values for equipment are A, B, and C



Answer to Practice SQL Statements (assuming the tables and PK constraints exist)





Set Constraints during Table Creation – Column Level

» Include in column definition

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for "Worksheet" and "Query Builder", with "Worksheet" selected. The main area displays the following SQL code:

```
CREATE TABLE DEPT
(deptid NUMBER(2) CONSTRAINT dept_deptid_pk PRIMARY KEY,
 dname VARCHAR2(20) NOT NULL
                      CONSTRAINT dept_dname_uk UNIQUE,
 fax VARCHAR(12)
);
```

The "deptid" column is defined with a primary key constraint ("CONSTRAINT dept_deptid_pk PRIMARY KEY"). The "dname" column is defined with a unique constraint ("CONSTRAINT dept_dname_uk UNIQUE"). The "fax" column is defined without any explicit constraints. Below the code, the "Script Output" tab is visible, showing the message "Task completed in 0.169 seconds".

Table DEPT created.

» Industry convention is to use Table-level constraints though...



Set Constraints during Table Creation – Table Level (1/2)

Worksheet Query Builder

```
CREATE TABLE dept
(deptid NUMBER(2),
 dname VARCHAR2(20) NOT NULL, → 2
 fax VARCHAR2(12),
 CONSTRAINT dept_deptid_pk PRIMARY KEY (deptid),
 CONSTRAINT dept_dname_uk UNIQUE (dname)
);

```



```
CREATE TABLE etypes
(etypeid NUMBER(3),
 etypename VARCHAR2(20) NOT NULL, → 4
 CONSTRAINT etypes_etypeid_pk PRIMARY KEY (etypeid),
 CONSTRAINT etypes_etypename_uk UNIQUE (etypename)
);
```

Query Result Script Output Task completed in 0.058 seconds

» Include the constraints at the end of the column list

» Create parent tables first!

Table DEPT created.

Table ETYPES created.



Set Constraints during Table Creation – Table Level (2/2)

Worksheet Query Builder

```
CREATE TABLE equip
(equipid NUMBER(3),
 edesc VARCHAR2(30),
 purchdate DATE,
 rating CHAR(1),
 deptid NUMBER(2) NOT NULL, → 5.2
 etypeid NUMBER(3),
CONSTRAINT equip_equipid_pk PRIMARY KEY (equipid),
CONSTRAINT equip_deptid_fk FOREIGN KEY (deptid) → 5.1
    REFERENCES dept (deptid),
CONSTRAINT equip_etypeid_fk FOREIGN KEY (etypeid) → 6
    REFERENCES etypes (etypeid),
CONSTRAINT equip_rating_ck CHECK (rating IN ('A', 'B', 'C')) → 7
);

```

Script Output X | Task completed in 0.049 seconds

Table EQUIP created.



Multiple Constraints on a Single Column

- » A column may be included in multiple constraints
- » The order# column is included in a primary key and a foreign key constraint

```
CREATE TABLE ORDERITEMS
(Order# NUMBER(4),
 Item# NUMBER(2),
 ISBN VARCHAR2(10),
 Quantity NUMBER(3) NOT NULL,
 PaidEach NUMBER(5,2) NOT NULL,
 CONSTRAINT orderitems_order#item#_pk PRIMARY KEY (order#, item#),
 CONSTRAINT orderitems_order#_fk FOREIGN KEY (order#)
    REFERENCES orders (order#),
 CONSTRAINT orderitems_isbn_fk FOREIGN KEY (isbn)
    REFERENCES books (isbn),
 CONSTRAINT orderitems_quantity_ck CHECK (quantity > 0) );
```



Viewing Constraints: USER_CONSTRAINTS

Worksheet Query Builder

```
SELECT constraint_name, constraint_type, search_condition, r_constraint_name  
FROM user_constraints  
WHERE table_name = 'EQUIP';
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 2.412 seconds

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_CONSTRAINT_NAME
1 EQUIP_DEPTID_FK	R	(null)	DEPT_DEPTID_PK
2 EQUIPETYPEID_FK	R	(null)	ETYPES_ETYPEID_PK
3 SYS_C0022140	C	"DEPTID" IS NOT NULL	(null)
4 EQUIP_RATING_CK	C	rating IN ('A', 'B', 'C')	(null)
5 EQUIPEQUIPID_PK	P	(null)	(null)

» Display constraint list for a specific table

C	Check or NOT NULL constraint
P	Primary Key constraint
R	Foreign Key constraint
U	Unique constraint

Display the condition in a CHECK constraint or NOT NULL

Display the name of the PK constraint on the referenced column



Viewing Constraints: USER_CONS_COLUMNS

- » Display constraint listing with column names

Worksheet Query Builder

```
SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'EQUIP';
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.023 seconds

CONSTRAINT_NAME	COLUMN_NAME
SYS_C0022140	DEPTID
EQUIP_RATING_CK	RATING
EQUIP_EQUIPID_PK	EQUIPID
EQUIP_DEPTID_FK	DEPTID
EQUIPETYPEID_FK	ETYPEID



Using DISABLE/ENABLE

- » Sometimes, we need to disable the constraints...
- » Use DISABLE or ENABLE clause of ALTER TABLE command

ALTER TABLE tablename

DISABLE CONSTRAINT constraintname;

ALTER TABLE tablename

ENABLE CONSTRAINT constraintname;

```
Worksheet Query Builder
```

```
ALTER TABLE equip  
DISABLE CONSTRAINT equip_rating_ck;
```

```
ALTER TABLE equip  
ENABLE CONSTRAINT equip_rating_ck;
```

```
Query Result Script Output
```

Task completed in 0.072 seconds

Table EQUIP altered.

Table EQUIP altered.



Dropping Constraints

- » Constraints cannot be modified; they must be dropped and recreated
- » Actual syntax depends on the type of constraint

PRIMARY KEY

– just list type of constraint

```
ALTER TABLE tablename
DROP PRIMARY KEY;
```

No column
name

UNIQUE

– include column name

```
ALTER TABLE tablename
DROP UNIQUE (columnname);
```

All others

– reference constraint name

```
ALTER TABLE tablename
DROP CONSTRAINT constraintname;
```



Drop Constraint Example

Worksheet Query Builder

```
ALTER TABLE equip
DROP CONSTRAINT equip_rating_ck;
```

Query Result x Script Output x

Task completed in 0.161 seconds

Table EQUIP altered.

```
constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'EQUIP';
```

Query Result x Script Output x

All Rows Fetched: 4 in 1.313 seconds

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	EQUIP_DEPTID_FK	R	(null)
2	EQUIPETYPEID_FK	R	(null)
3	SYS_C0022140	C	"DEPTID" IS NOT NULL
4	EQUIP_EQUIPID_PK	P	(null)

equip_rating_ck
constraint is removed



Drop Constraint Example – Error

Worksheet Query Builder

```
ALTER TABLE dept
DROP PRIMARY KEY;
```

Query Result Script Output Task completed in 0.076 seconds

Error starting at line : 355 in command -
ALTER TABLE dept
DROP PRIMARY KEY
Error report -
ORA-02273: this unique/primary key is referenced by some foreign keys
02273. 00000 - "this unique/primary key is referenced by some foreign keys"
*Cause: Self-evident.
*Action: Remove all references to the key before the key is to be dropped.

Drop Primary Key Constraint



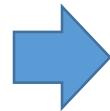
» **CASCADE** option will remove the primary key of a parent table

Worksheet Query Builder

```
ALTER TABLE dept
DROP PRIMARY KEY CASCADE;
```

Query Result x Script Output x

Task completed in 0.086 seconds



Removes a Foreign key constraint
of its child table first, then removes
its PK constraint

Table DEPT altered.

Worksheet Query Builder

```
SELECT constraint_name, constraint_type, search_condition
FROM user_constraints
WHERE table_name = 'EQUIP';
```

Script Output x Query Result x

All Rows Fetched: 3 in 0.042 seconds

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 EQUIPETYPEID_FK	R	(null)
2 SYS_C0022140	C	"DEPTID" IS NOT NULL
3 EQUIPEQUIPID_PK	P	(null)

equip_deptid_fk
constraint is removed





Check the Child Table's FK Constraint

Worksheet Query Builder

```
SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'EQUIP';
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.231 seconds

	CONSTRAINT_NAME	COLUMN_NAME
1	SYS_C0012901	DEPTID
2	EQUIP_RATING_CK	RATING
3	EQUIP_EQUIPID_PK	EQUIPID
4	EQUIP_DEPTID_FK	DEPTID
5	EQUIPETYPEID_FK	ETYPEID

→ eliminated (removed in slide 49)

→ eliminated (auto-removed when PK of its parent table 'DEPT' is dropped with CASCADE)



Summary (1/3)

- » A constraint is a rule that is applied to data being added to a table
 - The constraint represents business rules, policies, and/or procedures
 - Data violating the constraint is not added to the table
- » A constraint can be included during table creation as part of the CREATE TABLE command or added to an existing table by using the ALTER TABLE command



Summary (2/3)

- » A PRIMARY KEY constraint does not allow duplicate or NULL values in the designated column
- » Only **ONE** PRIMARY KEY constraint is allowed in a table
- » A FOREIGN KEY constraint ensures that the column entry match a referenced column entry in the referenced table (Parent table) or be NULL
- » A UNIQUE constraint is similar to a PRIMARY KEY constraint, but it allows NULL values to be stored in the specified column
- » A CHECK constraint ensures a value meets a specified condition
- » A NOT NULL constraint ensures a value must be provided for a column



Summary (3/3)

- » A constraint can be disabled or enabled using the ALTER TABLE command with the DISABLE and ENABLE keywords
- » A constraint cannot be modified. To change a constraint, the constraint must first be dropped with the DROP command and then re-created
- » USER_CONSTRAINTS and USER_CONS_COLUMNS data dictionary views provide information regarding constraints



Let's Check Our Knowledge!

The Kahoot! logo is displayed in white text against a dark purple background. The word "Kahoot!" is written in a bold, sans-serif font, with an exclamation mark at the end. A large, semi-transparent purple arrow shape points diagonally upwards from the bottom left towards the logo.