

Creating Tables

Sub-topics

- Table Creation
- Creating a Table
- Data Integrity and Table Constraints
- Manage, Drop, and Alter Tables
- Relating Tables – Identifying Foreign Keys
- Maintaining Referential Integrity
- Composite Primary Keys and Multiple Foreign Keys
- Populating Tables with Data

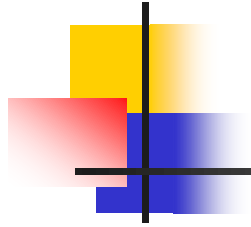


Table Creation

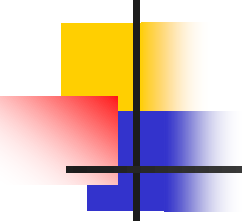
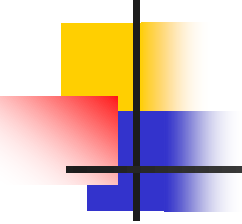


Table Creation

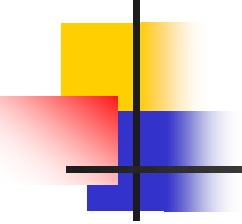
To create a table, 4 pieces of information must be determined:

1. Table name
2. Column names
3. Column data types
4. Column sizes



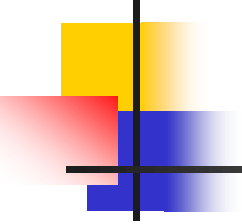
Predefined Data Types

- CHAR (*n*) - fixed-length character column
 - maximum length is 255
 - default length is 1 character
- VARCHAR (*n*) - variable-length character data
 - maximum length is 65,535
 - no default size



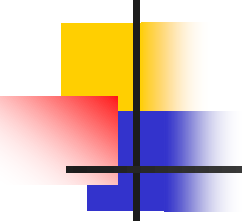
Predefined Data Types (continued)

- DECIMAL (p, s) – used for floating/real numbers, where p (precision) is the total no. of digits to the left & right of the decimal point; and s (scale) is the no. of positions to the right of the decimal point.



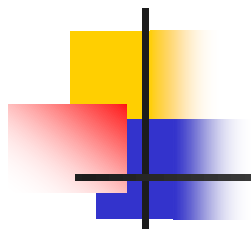
Predefined Data Types (continued)

- INT - stores integer values (whole numbers)
- DATE - stores date and time
 - default format is YYYY-MM-DD

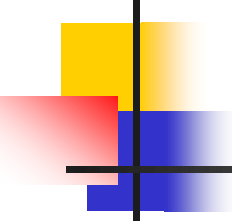


Creating a Table (example)

```
create table employee
(
    empID            char(9),
    lastName          varchar(15),
    firstName         varchar(15),
    birthDate         date,
    salary            decimal(7,2),
    parkingSpace      int
);
```

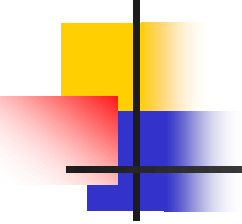


Data Integrity and Table Constraints



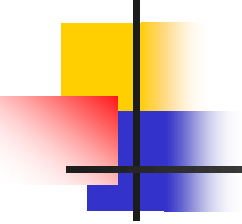
CONSTRAINTS : Definition

Rules used to enforce business rules, practices, and policies – and to ensure the accuracy and integrity of data.



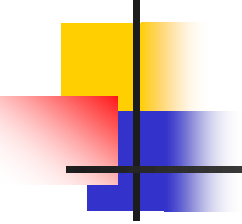
Constraint Types

- PRIMARY KEY - determines which column(s) uniquely identifies each row
 - can't be NULL and values must be unique
- FOREIGN KEY - ensures that if a value is entered into a specified column of the child table, it must already exist in the parent table



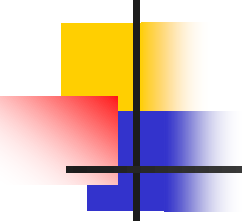
Constraint Types (continued)

- **UNIQUE** - ensures that all data values stored in a specified column are unique
 - allows NULL values
- **CHECK** - ensures that a specified condition is true before the data value is added to a table



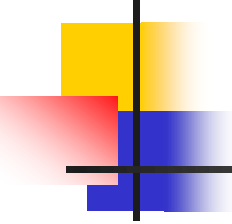
Constraint Types (continued)

- **DEFAULT** - used to set a default value for a specified column
- **NOT NULL** - ensures that a specified column cannot contain a NULL value

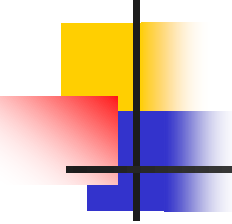


Updated CREATE TABLE command

```
create table employee  
(empID char(9) primary key,  
lastName varchar(15) not null,  
firstName varchar(15) not null,  
birthDate date,  
salary decimal(7,2) not null check (salary<=50000),  
parkingSpace int not null unique  
);
```

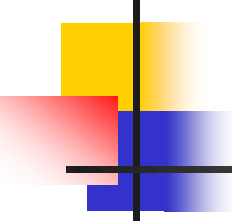


Manage, Drop, and Alter Tables



Viewing a Table Structure (example)

```
DESCRIBE employee;
```

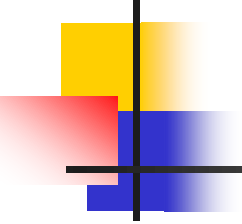


Dropping/Deleting a Table : Syntax

`DROP TABLE tablename;`

Dropping a Table (example)

`DROP TABLE employee;`

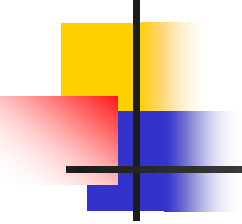


Renaming a Table : Syntax

RENAME TABLE *oldtablename* TO *newtablename*;

Renaming a Table (example)

RENAME TABLE employee TO worker;

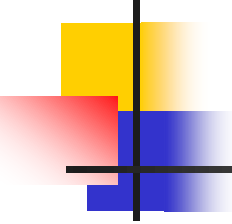


Dropping a Column (example)

```
ALTER TABLE employee  
DROP COLUMN bDate;
```

Changing a Column Name (example)

```
ALTER TABLE employee CHANGE [COLUMN]  
lastName last_name varchar(15) not null;
```

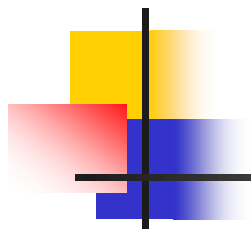


Altering a Column (example)

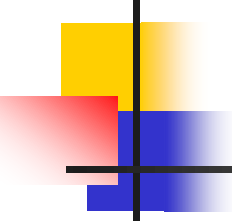
```
ALTER TABLE employee MODIFY  
firstName varchar(20) not null;
```

Adding a Column (example)

```
ALTER TABLE employee ADD  
gender char;
```

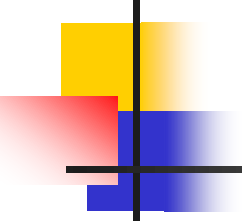


Relating Tables - Identifying Foreign Keys



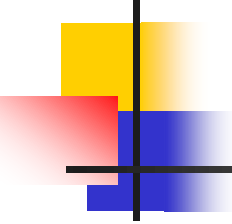
Creating Another Table (example)

```
CREATE TABLE department  
( depCode char(2) PRIMARY KEY,  
  depName varchar(30) NOT NULL  
);
```



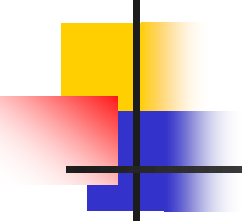
Creating a Table w/ a Foreign Key (example)

```
create table employee
(empID char(9) primary key,
 lName varchar(15) not null,
 fName varchar(15) ) not null,
 bDate date,
 salary decimal(7,2) ) not null check (salary<=50000),
 parkingSpace int not null unique,
 depCode char(2),
 foreign key (depCode) references department (depCode)
 on delete set null on update cascade
);
```



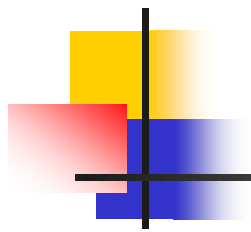
Adding a Foreign Key to an Existing Table

```
ALTER TABLE employee ADD  
( depCode char(2), FOREIGN KEY  
(depCode) REFERENCES department (depCode)  
ON DELETE SET NULL ON UPDATE CASCADE);
```

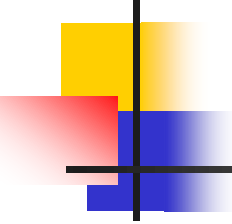


Additional Referential Integrity Constraints

- ON UPDATE CASCADE | SET NULL | RESTRICT
- ON DELETE CASCADE | SET NULL | RESTRICT

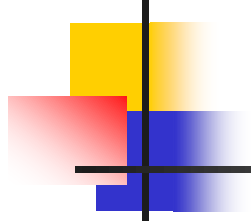


Composite Primary Keys and Multiple Foreign Keys

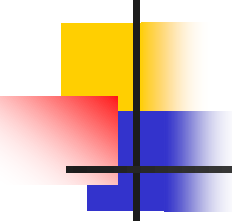


Composite Primary Keys and Multiple Foreign Keys (example)

```
create table assignment
( empID char(9),
  projNum int,
  hours int,
  primary key (empID, projNum),
  foreign key (empID) references employee (empID)
  on delete restrict on update cascade,
  foreign key (projNum) references project (projNum)
  on delete restrict on update cascade
);
```



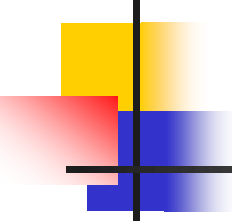
Populating Tables with Data



The INSERT Command

General Form:

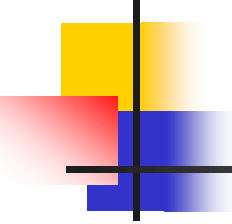
```
INSERT INTO table-name  
[(column1-name, column2-name, ...)]  
VALUES (column1-value, column2-value, ...);
```



The INSERT Command (examples)

```
INSERT INTO department VALUES (12, 'Production');
```

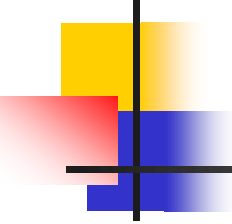
```
INSERT INTO employee (empID, lName, fName, salary)  
VALUES ('123456789', 'Smith', 'John', 25000);
```



The DELETE Command

General Form:

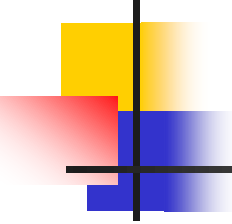
```
DELETE FROM table-name  
[WHERE condition];
```



The DELETE Command (examples)

```
DELETE FROM employee  
WHERE salary >= 5000 and salary <= 10000;
```

```
DELETE FROM employee;
```



The UPDATE Command

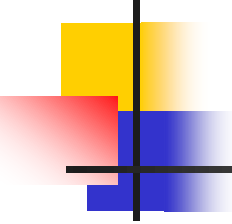
General Form:

UPDATE table-name

SET column1-name = expression

[, column2-name = expression] . . .

[WHERE condition];



The UPDATE Command (examples)

```
UPDATE employee  
SET parkingSpace = 253  
WHERE empID = '123456789';
```

```
UPDATE employee  
SET salary = salary * 1.05;
```