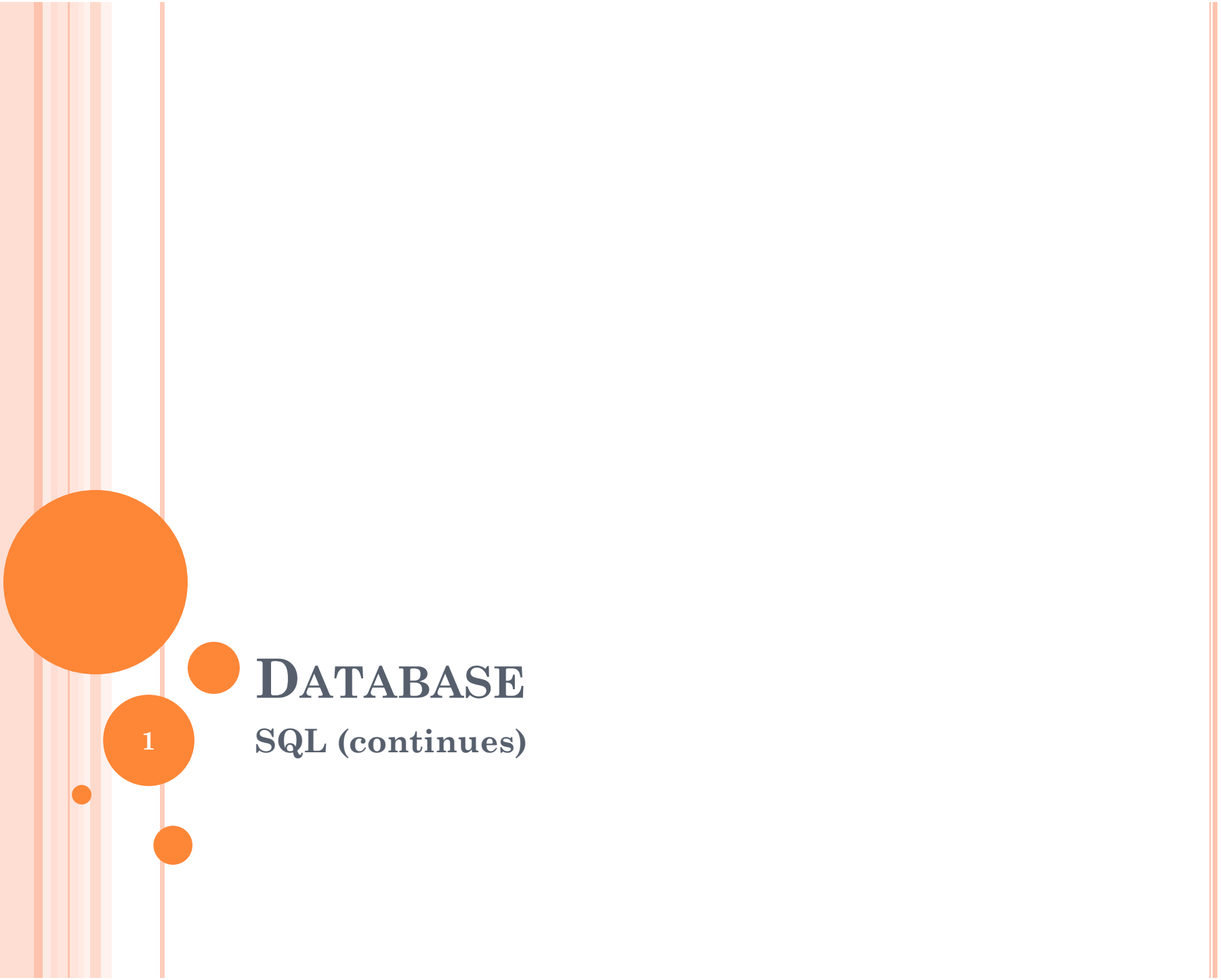

DATABASE

SQL (continues)

SOME MORE USEFUL COMMANDS

- At first, we will learn the commands to do the followings-
  - How to create a backup of the database?
  - How to encrypt and decrypt the backup file?
  - How to use view?
  - Some string functions

MYSQLDUMP

- This utility can be used to create sql format dump file or backup file
  - $mysqldump [arguments] > fileName
- To dump a single database
  - $mysqldump –u username –p db1> dump.sql
- To dump specific tables of database
  - $mysqldump –u username –p db1 t1 t2 t3> dump.sql

RELOAD THE SQL FORMAT DUMP FILES

- To reload a dump file written by mysqldump that consists of SQL statements, use the followings
  - $ mysql –u username –p db1 < dump.sql
- From within mysql,
  - mysql>use db1
  - mysql> source dump.sql

Encrypting and Decrypting the dump file

- We can use openssl command
- Encryption
  - If we want to encrypt test.sql file using openssl then we can use the following
  - $openssl aes-256-cbc –a –salt -in test.sql –out testop.txt
  - This will prompt you to use a password. Once you verify the password then the file will be encrypted
- Decryption
  - $openssl aes-256-cbc –d –a -in testop.txt –out testnew.sql
  - This will prompt you to enter a password. Once you enter the password used for encryption then the file will be decrypted

SOME STRING FUNCTIONS

- ASCII
  - Returns the ASCII value of a character
  - mysql> SELECT ASCII('a');
- Char_length
  - Returns the length of the string in characters
  - mysql> SELECT CHAR_LENGTH('university');
- Concat
  - Adds two or more expressions together
  - mysql> SELECT CONCAT('delhi ' ,'university');
- Concat_ws
  - Adds two or more expressions together with separator
  - mysql> SELECT CONCAT_WS('-', 'delhi ','university');

SOME STRING FUNCTIONS

- Field
  - Returns the index position of a value in a list of values
  - Syntax: FIELD(*value*, *val1*, *val2*, *val3*, ...)
  - mysql> SELECT FIELD ('a','k','d','s','a');
- Format
  - The FORMAT() function formats a number to a format like "#,###,###.##", rounded to a specified number of decimal places, then it returns the result as a string.
  - Syntax: FORMAT(*number*, *decimal_places*)
  - mysql> SELECT FORMAT(123456789.789,2)

SOME STRING FUNCTIONS

- Insert()
  - The INSERT() function inserts a string within a string at the specified position and for a certain number of characters.
  - Syntax: INSERT(*string*, *position*, *number*, *string2*)
  - mysql> SELECT INSERT("univ.edu.in", 6, 3 "gov");
- Instr()
  - The INSTR() function returns the position of the first occurrence of a string in another string
  - Syntax: INSTR(*string*, *substring*)
  - mysql>
    SELECT INSTR("du.edu", "edu") AS MatchPosition;

SOME STRING FUNCTIONS

- Lcase() or Lower()
  - Converts a string to lowercase
  - Syntax: Lcase(text), Lower(text)
  - mysql>SELECT LCASE('UNIVERSITY')
  - mysql>SELECT LOWER('UNIVERSITY')
- Ucase or Upper()
  - Converts a string to uppercase
  - Syntax: Ucase(text), Upper(text)
  - mysql>SELECT UCASE('University')
  - mysql>SELECT UPPER('University')

SOME MORE STRING FUNCTIONS

- LOCATE()
  - The LOCATE() function returns the position of the first occurrence of a substring in a string.
  - Syntax: LOCATE(*substring*, *string*, *start*)
  - mysql>
    SELECT LOCATE("com", “www.yahoo.com", 4) AS M
    atchPosition;

- LPAD()
  - Left-pads a string with another string, to a certain length
  - Syntax: LPAD(*string*, *length*, *lpad_string*)
  - mysql>
    SELECT LPAD(CustomerName, 30, “*") AS LeftPad
    CustomerName FROM Customers;
- RPAD()
  - Similar to LPAD() but it right-pads a string with another string, to a certain length

SOME MORE STRING FUNCTIONS

- LTRIM()
  - The LTRIM() function removes leading spaces from a string
  - Syntax: LTRIM(*string*)
  - mysql> SELECT LTRIM(‘ delhi’);
- RTRIM()
  - Similar to LTRIM. The RTRIM() function removes trailing spaces from a string

SOME MORE STRING FUNCTIONS

- REPEAT()
  - The REPEAT() function repeats a string for a specified number of times
  - Syntax: REPEAT(*string*, *number*)
  - mysql> SELECT REPEAT('abc',3);
- Replace
  - The REPLACE() function replaces all occurrences of a substring within a string, with a new substring
  - Syntax:
    REPLACE(*string*, *from_substring*, *new_substring*)
  - mysql> SELECT REPLACE("SQL Tutorial", "SQL", "HTML");

SOME MORE STRING FUNCTIONS

- REVERSE()
  - The REVERSE() function reverses a string and returns the result.
  - Syntax: reverse(string)
  - mysql> SELECT REVERSE('delhi');

SOME MORE STRING FUNCTIONS

- STRCMP()
  - The STRCMP() function compares two strings
  - Syntax: strcmp(string1, string2))
  - If *string1* = *string2*, this function returns 0
  - If *string1* < *string2*, this function returns -1
  - If *string1* > *string2*, this function returns 1
  - mysql> SELECT STRCMP('abc','abb');

SOME MORE STRING FUNCTIONS

- SUBSTR()
  - The SUBSTR() function extracts a substring from a string (starting at any position).
  - Syntax: SUBSTR(*string*, *start*, *length*)
  - mysql>
    SELECT SUBSTR(CustomerName, 2, 5) AS ExtractString FROM Customers;

- RIGHT() Function
  - Extracts a specified number of characters from the right side of a string
  - Syntax: RIGHT(str, len)
  - mysql> SELECT RIGHT('IIT Patna',5)

VIEW

- A **view** consists of a stored query accessible as a *virtual table* composed of the result set of a query
- Unlike ordinary tables in a relational database, a view does not form part of the physical schema
- It is a dynamic, virtual table computed from data in the database
- Changing the data in a table alters the data shown in subsequent invocations of the view

VIEW

- A view is a *virtual table* that contains no physical data. It provides an alternative way to look at the data.
  - mysql> CREATE VIEW cust_view AS SELECT customer_name AS cust_name, customer_city AS cust_city FROM customer;
- Once a view is created, it can be used like a table
  - mysql> select * from cust_view;

USES OF VIEWS

- Hiding some information from some users
  - Consider a user who needs to know a customer's name, loan number and branch name, but has no need to see the loan amount.
  - mysql> **CREATE VIEW** *cust_loan_data* **AS**
    SELECT *customer_name, borrower.loan_number, branch_name*
    FROM *borrower, loan*
    WHERE *borrower.loan_number = loan.loan_number*
  - Grant the user permission to read *cust_loan_data*, but not *borrower or loan*

USES OF VIEW (2)

- Views can act as aggregated tables, where the database engine aggregates data (sum, average etc.) and presents the calculated results as part of the data
  - mysql>CREATE VIEW Category_Sales_For_1997 AS
    SELECT DISTINCT CategoryName, Count(Item), Sum(ProductSales)
    FROM Product, Sales
    WHERE Product.Id=Sales.Item.Id and Sales_year = 1997
    GROUP BY CategoryName