Logical database design for hr management system

Technology, Computer

Task 1. 1 The background information of the organization and operation that would support.

In an organization a HR department is responsible for record each employee. Where the employees have an identification number, job identification code, e-mail address, manager as well as salary. They also track those employees earn incentive or commissions in addition to their salary.

However, the company also tracks their role in the organization. Each job also recorded according to the characteristics. Moreover, ever jobs have job title, identification code, maximum and minimum salary of the job. There are few employees work for a long time with the company and they have held different department within the company. If any employee resigns, then the job identification number and department are recorded. The company also track the location of its departments and warehouses. Every employee must assign with a department where departments are identified by the unique identification number. Those departments are associated with different locations. The company need to store the location such as the state, city, postal code, street name as well as county code. The company also record the county name, currency name and the region.

This database supports a better employee management plan as well as their departments, location and associated jobs. However, the company would have a better structure to store their confidential information. This database will provide a better extracted information to developed their insufficiency. This efficient data structure allows them increases their storage as well as it exclude the redundancy in data.

Task 1. 2 a conceptual database design and list of enterprise rules

Figure 1: EER-diagram showing all enterprise rules

(Source: Created by author)

Task2. 1: A Logical Database Design for HR management System

Figure 2: logical database design

(Source: Created by author)

Task2. 2: Create the tables using Oracle DBMS

— ——————————-

— Table structure for COUNTRIES

— ——————————-

DROP TABLE “ MYDB”.” COUNTRIES”;

CREATE TABLE “ MYDB”.” COUNTRIES” (

“ country_id” VARCHAR2(30 BYTE) NOT NULL ,

“ country_name” VARCHAR2(30 BYTE) NULL ,

“ region_id” VARCHAR2(30 BYTE) NULL

)

LOGGING

```
NOCOMPRESS

NOCACHE

;

— ——————————-

— Table structure for DEPARTMENTS

— ——————————-

DROP TABLE “ MYDB”.” DEPARTMENTS”;

CREATE TABLE “ MYDB”.” DEPARTMENTS” (

“ department_id” VARCHAR2(30 BYTE) NOT NULL ,

“ department_name” VARCHAR2(30 BYTE) NULL ,

“ manager_id” VARCHAR2(30 BYTE) NULL ,

“ location_id” VARCHAR2(30 BYTE) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;
```

— ——————————-

— Table structure for EMPLOYEES

— ——————————-

DROP TABLE “ MYDB”.” EMPLOYEES”;

CREATE TABLE “ MYDB”.” EMPLOYEES” (

“ employee_id” VARCHAR2(30 BYTE) NOT NULL ,

“ first_name” VARCHAR2(30 BYTE) NULL ,

“ last_name” VARCHAR2(30 BYTE) NULL ,

“ email” VARCHAR2(30 BYTE) NULL ,

“ phone_number” NUMBER(12) NULL ,

“ hire_date” DATE NULL ,

“ job_id” VARCHAR2(30 BYTE) NULL ,

“ salary” NUMBER(10, 2) NULL ,

“ commission” NUMBER(10, 2) NULL ,

“ manager_id” VARCHAR2(30 BYTE) NULL ,

“ department_id” VARCHAR2(30 BYTE) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;

— ——————————-

— Table structure for JOB_HISTORY

— ——————————-

DROP TABLE “ MYDB”.” JOB_HISTORY”;

CREATE TABLE “ MYDB”.” JOB_HISTORY” (

“ employee_id” VARCHAR2(30 BYTE) NOT NULL ,

“ start_date” DATE NULL ,

“ end_date” DATE NULL ,

“ job_id” VARCHAR2(30 BYTE) NULL ,

“ department_id” VARCHAR2(30 BYTE) NOT NULL

)

LOGGING

NOCOMPRESS

```
NOCACHE

;

— ——————————-

— Table structure for JOBS

— ——————————-

DROP TABLE “ MYDB”.” JOBS”;

CREATE TABLE “ MYDB”.” JOBS” (

“ job_id” VARCHAR2(30 BYTE) NOT NULL ,

“ job_title” VARCHAR2(30 BYTE) NULL ,

“ min_salary” NUMBER(10, 2) NULL ,

“ max_salary” NUMBER(10, 2) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;

— ——————————-
```

— Table structure for LOCATIONS

— ——————————-

DROP TABLE “ MYDB”.” LOCATIONS”;

CREATE TABLE “ MYDB”.” LOCATIONS” (

“ location_id” VARCHAR2(30 BYTE) NOT NULL ,

“ street_address” VARCHAR2(30 BYTE) NULL ,

“ postal_code” NUMBER(10) NULL ,

“ city” VARCHAR2(30 BYTE) NULL ,

“ state” VARCHAR2(30 BYTE) NULL ,

“ country_id” VARCHAR2(30 BYTE) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;

— ——————————-

— Table structure for REGIONS

— ——————————-

DROP TABLE “ MYDB”.” REGIONS”;

CREATE TABLE “ MYDB”.” REGIONS” (

“ region_id” VARCHAR2(30 BYTE) NOT NULL ,

“ region_name” VARCHAR2(30 BYTE) NULL

)

LOGGING

NOCOMPRESS

NOCACHE

;

Task2. 3: Create the four most useful indexes

— ——————————-

— Indexes structure for table COUNTRIES

— ——————————-

— ——————————-

— Checks structure for table COUNTRIES

— ——————————-

ALTER TABLE “ MYDB”.” COUNTRIES” ADD CHECK (“ country_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table COUNTRIES

— ——————————-

ALTER TABLE “ MYDB”.” COUNTRIES” ADD PRIMARY KEY (“ country_id”);

— ——————————-

— Indexes structure for table DEPARTMENTS

— ——————————-

— ——————————-

— Checks structure for table DEPARTMENTS

— ——————————-

ALTER TABLE “ MYDB”.” DEPARTMENTS” ADD CHECK (“ department_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table DEPARTMENTS

— ——————————-

ALTER TABLE “ MYDB”.” DEPARTMENTS” ADD PRIMARY KEY (“ department_id”);

— ——————————-

— Indexes structure for table EMPLOYEES

— ——————————-

— ——————————-

— Checks structure for table EMPLOYEES

— ——————————-

ALTER TABLE “ MYDB”.” EMPLOYEES” ADD CHECK (“ employee_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table EMPLOYEES

— ——————————-

ALTER TABLE “ MYDB”.” EMPLOYEES” ADD PRIMARY KEY (“ employee_id”);

— ——————————-

— Indexes structure for table JOB_HISTORY

— ——————————-

— ——————————-

— Checks structure for table JOB_HISTORY

— ——————————-

ALTER TABLE “ MYDB”.” JOB_HISTORY” ADD CHECK (“ employee_id” IS NOT NULL);

ALTER TABLE “ MYDB”.” JOB_HISTORY” ADD CHECK (“ department_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table JOB_HISTORY

— ——————————-

ALTER TABLE “ MYDB”.” JOB_HISTORY” ADD PRIMARY KEY (“ employee_id”);

— ——————————-

— Indexes structure for table JOBS

— ——————————-

— ——————————-

— Checks structure for table JOBS

— ——————————-

ALTER TABLE “ MYDB”.” JOBS” ADD CHECK (“ job_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table JOBS

— ——————————-

ALTER TABLE “ MYDB”.” JOBS” ADD PRIMARY KEY (“ job_id”);

— ——————————-

— Indexes structure for table LOCATIONS

— ——————————-

— ——————————-

— Checks structure for table LOCATIONS

— ——————————-

ALTER TABLE “ MYDB”.” LOCATIONS” ADD CHECK (“ location_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table LOCATIONS

— ——————————-

ALTER TABLE “ MYDB”.” LOCATIONS” ADD PRIMARY KEY (“ location_id”);

— ——————————-

— Indexes structure for table REGIONS

— ——————————-

— ——————————-

— Checks structure for table REGIONS

— ——————————-

ALTER TABLE “ MYDB”.” REGIONS” ADD CHECK (“ region_id” IS NOT NULL);

— ——————————-

— Primary Key structure for table REGIONS

— ——————————-

ALTER TABLE “ MYDB”.” REGIONS” ADD PRIMARY KEY (“ region_id”);

— ——————————-

— Foreign Key structure for table “ MYDB”.” COUNTRIES”

— ——————————-

ALTER TABLE “ MYDB”.” COUNTRIES” ADD FOREIGN KEY (“ region_id”)

REFERENCES “ MYDB”.” REGIONS” (“ region_id”) ON DELETE CASCADE;

— ——————————-

— Foreign Key structure for table “ MYDB”.” DEPARTMENTS”

— ——————————-

ALTER TABLE “ MYDB”.” DEPARTMENTS” ADD FOREIGN KEY (“ location_id”) REFERENCES “ MYDB”.” LOCATIONS” (“ location_id”) ON DELETE CASCADE;

— ——————————-

— Foreign Key structure for table “ MYDB”.” EMPLOYEES”

— ——————————-

ALTER TABLE “ MYDB”.” EMPLOYEES” ADD FOREIGN KEY (“ job_id”) REFERENCES “ MYDB”.” JOBS” (“ job_id”) ON DELETE CASCADE;

ALTER TABLE “ MYDB”.” EMPLOYEES” ADD FOREIGN KEY (“ department_id”) REFERENCES “ MYDB”.” DEPARTMENTS” (“ department_id”) ON DELETE CASCADE;

— ——————————-

— Foreign Key structure for table “ MYDB”.” JOB_HISTORY”

— ——————————-

ALTER TABLE “ MYDB”.” JOB_HISTORY” ADD FOREIGN KEY (“ employee_id”) REFERENCES “ MYDB”.” EMPLOYEES” (“ employee_id”) ON DELETE CASCADE;

— ——————————-

— Foreign Key structure for table “ MYDB”.” LOCATIONS”

— ——————————-

ALTER TABLE “ MYDB”.” LOCATIONS” ADD FOREIGN KEY (“ country_id”)

REFERENCES “ MYDB”.” COUNTRIES” (“ country_id”) ON DELETE CASCADE;

Task2. 4: Data Population

The below figures showing all data in each table:

Table countries:

Table departments:

Table employees:

Table job_history:

Table jobs:

Table locations:

Table regions:

Task2. 5: SQL Query writing

Query 1

SELECT

MYDB. COUNTRIES.” country_name”

FROM

MYDB. COUNTRIES

Query 2

SELECT

MYDB. REGIONS.” region_name”,

MYDB. COUNTRIES.” country_name”

FROM

MYDB. COUNTRIES

INNER JOIN MYDB. REGIONS ON MYDB. COUNTRIES.” region_id” = MYDB. REGIONS.” region_id”

Query 3

SELECT

MYDB. JOB_HISTORY.” start_date”,

MYDB. JOB_HISTORY.” end_date”,

MYDB. EMPLOYEES.” first_name”,

MYDB. EMPLOYEES.” last_name”,

MYDB. EMPLOYEES.” email”

FROM

MYDB. EMPLOYEES

FULL OUTER JOIN MYDB. JOB_HISTORY ON MYDB. JOB_HISTORY.”

employee_id” = MYDB. EMPLOYEES.” employee_id”

Query 4

SELECT

Count(MYDB. EMPLOYEES.” employee_id”) AS “ Number Of Employee”

FROM

MYDB. EMPLOYEES

Query 5

SELECT

MYDB. EMPLOYEES.” first_name”,

MYDB. EMPLOYEES.” last_name”,

MYDB. EMPLOYEES.” email”,

MYDB. EMPLOYEES.” phone_number”,

MYDB. EMPLOYEES.” hire_date”,

MYDB. EMPLOYEES.” salary”,

MYDB. EMPLOYEES.” commission”

FROM

MYDB. EMPLOYEES

ORDER BY

MYDB. EMPLOYEES.” first_name” ASC

Query 6

SELECT

MYDB. EMPLOYEES.” first_name”,

MYDB. EMPLOYEES.” last_name”,

MYDB. EMPLOYEES.” email”,

MYDB. EMPLOYEES.” phone_number”,

MYDB. EMPLOYEES.” hire_date”,

MYDB. EMPLOYEES.” salary”,

MYDB. EMPLOYEES.” commission”

FROM

MYDB. EMPLOYEES

WHERE

MYDB. EMPLOYEES.” email” LIKE ‘%gmail%’

Query 7

SELECT

MYDB. EMPLOYEES.” first_name”,

MYDB. EMPLOYEES.” last_name”,

MYDB. EMPLOYEES.” email”,

MYDB. EMPLOYEES.” phone_number”

FROM

MYDB. EMPLOYEES

INNER JOIN MYDB. JOB_HISTORY ON MYDB. JOB_HISTORY.” employee_id” =

MYDB. EMPLOYEES.” employee_id”

WHERE

MYDB. JOB_HISTORY.” employee_id” IN (MYDB. EMPLOYEES.” employee_id”)

Query 8

MYDB. EMPLOYEES.” email”,

MYDB. EMPLOYEES.” phone_number”,

MYDB. EMPLOYEES.” hire_date”,

MYDB. EMPLOYEES.” job_id”,

MYDB. EMPLOYEES.” salary”,

MYDB. EMPLOYEES.” commission”,

MYDB. EMPLOYEES.” manager_id”,

MYDB. EMPLOYEES.” department_id”,

MYDB. EMPLOYEES.” employee_id”

FROM

MYDB. EMPLOYEES,

(SELECT MYDB. JOB_HISTORY.” employee_id” fromA, A MYDB. JOB_HISTORY) subquery1

WHERE

subquery1.” employee_id”= MYDB. EMPLOYEES.” employee_id”

A

Asabe, S. A., Oye, N. D. and Goji, M., 2013. Hospital patient database management system: A case study of general hospital north-bank makurdi-nigeria. *Compusoft* , (3), p. 65.

Coronel, C. and Morris, S., 2016. *Database systems: design, implementation, & management* . Cengage Learning.

Dorok, S., BreAY, S., Teubner, J. and Saake, G., 2015. Flexible Analysis of Plant Genomes in a Database Management System. In *EDBT* (pp. 509-512).

Hussain, M., Pandey, A. C. and Pachauri, S., 2013. Performanc Tuning of Database Management System by Fuzzy Controlled Architecture. *Pragyaan: Journal of Information Technology* , p. 30.

Jahn, M., Schill, E. and Breunig, M., 2013. Towards a 4D database management system for geothermal projects: an example of the hydraulic data of Soultz. In *Second European Geothermal Workshop* .

Lee, H., Chapiro, J., Schernthaner, R., Duran, R., Wang, Z., Gorodetski, B., Geschwind, J. F. and Lin, M., 2015. How I do it: a practical database management system to assist clinical research teams with data collection, organization, and reporting. *Academic radiology* , (4), pp. 527-533.

Li, Z. and Shen, H., 2016. Database Design on Teaching Management System Based on SQL Server.

Mohamed, A. R., Kumar, P. V., Abhilash, S., Ravishankar, C. N. and Edwin, L., 2013. Design and Development of an Online Database Management System (AGRI-TECHBASE): For Agricultural Technologies of ICAR. In *Driving the Economy through Innovation and Entrepreneurship* (pp. 869-877). Springer India.

Nidzwetzki, J. K. and GA? ting, R. H., 2016. DISTRIBUTED SECONDO: An extensible highly available and scalable database management system.

Reddy, T. B. K., Thomas, A. D., Stamatis, D., Bertsch, J., Isbandi, M., Jansson, J., Mallajosyula, J., Pagani, I., Lobos, E. A. and Kyrpides, N. C., 2014. The Genomes OnLine Database (GOLD) v. 5: a metadata management system based on a four level (meta) genome project classification. *Nucleic acids research* , p. gku950.

Sui, X. L., Wang, D., Liu, X. Y. and Teng, Y., 2014. Database Design of NC Cutting Tool Matching and Management System. In *Advanced Materials Research* (Vol. 981, pp. 546-550). Trans Tech Publications.