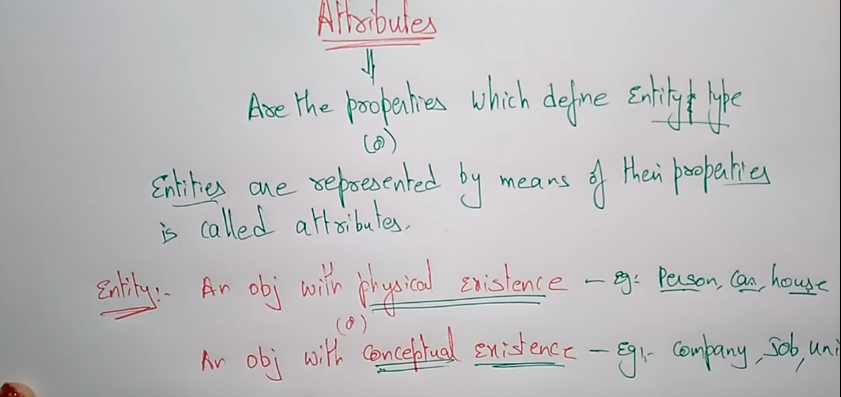
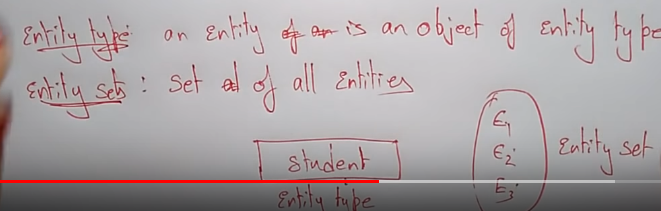
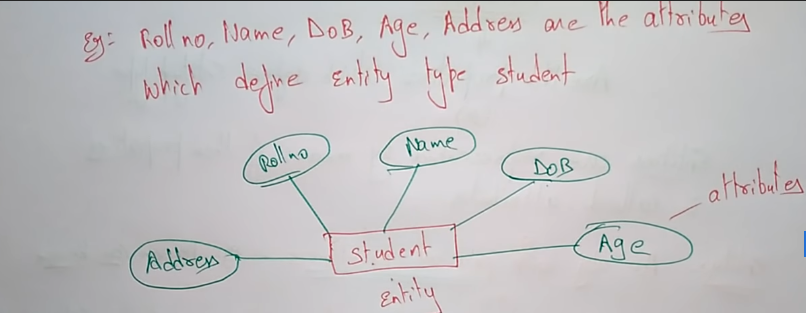
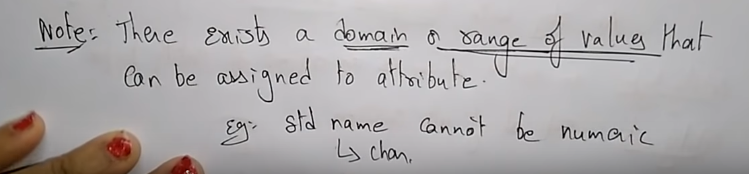
 

Attribute, entity

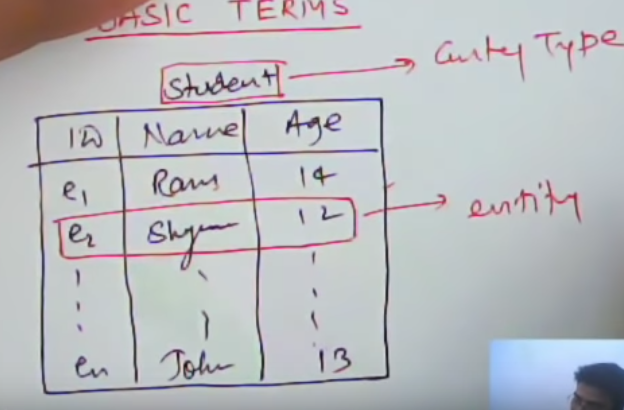
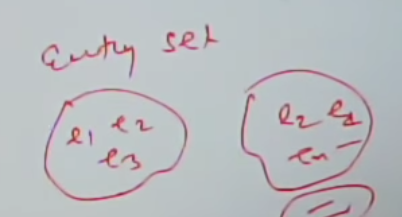




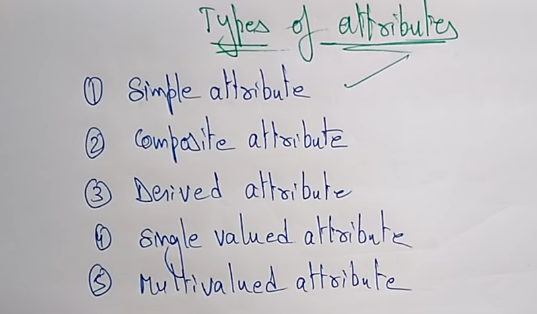


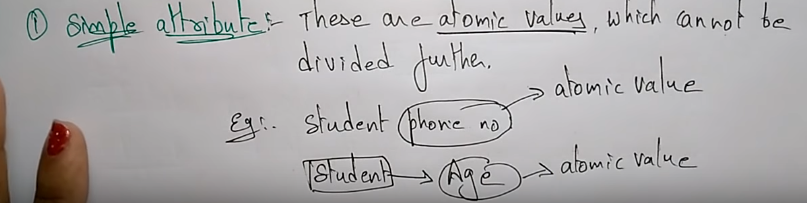


Each row is a entity and table name is entity type and collection of entity is entity set

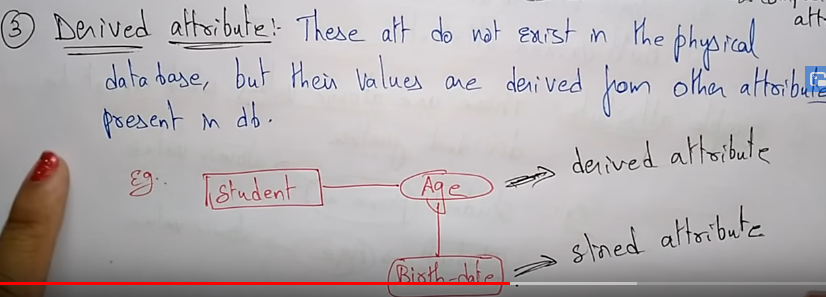
 

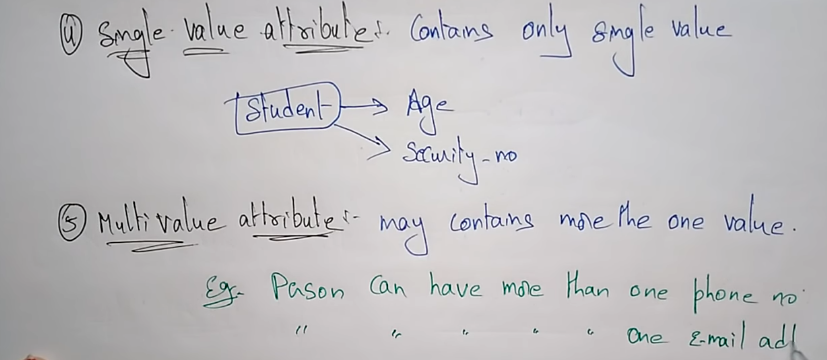
**Attribute** -🡪 id, name , age are attribute and ram,sam, 11,12, e1,e2 are property

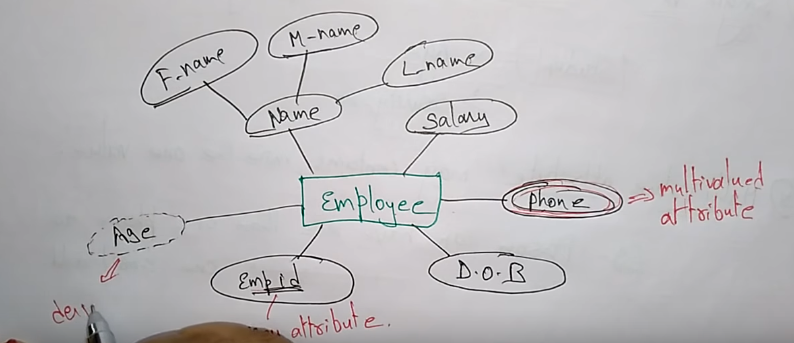


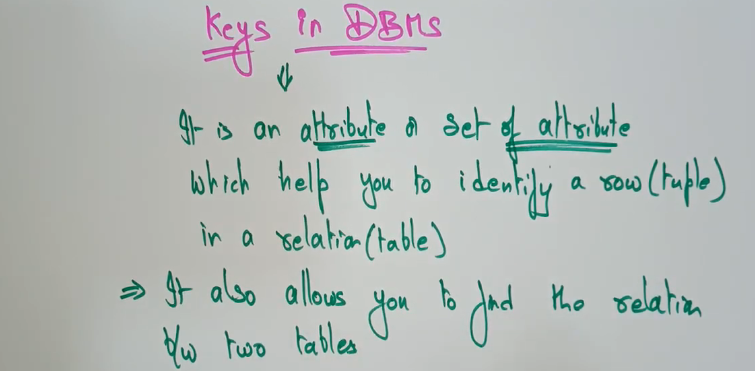




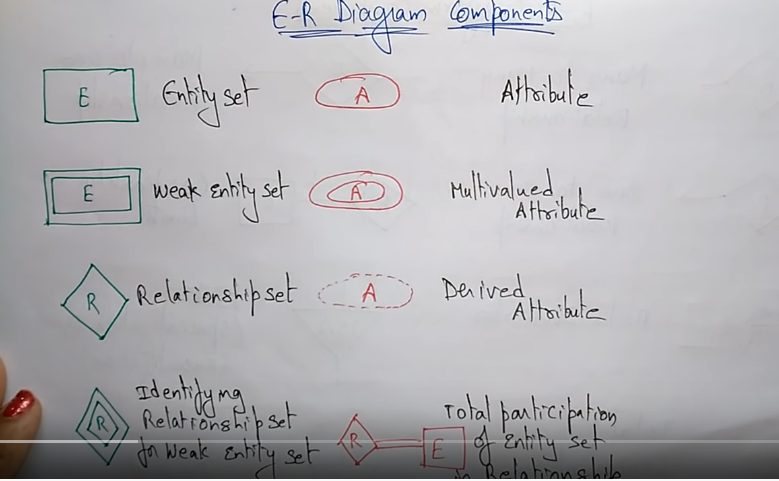


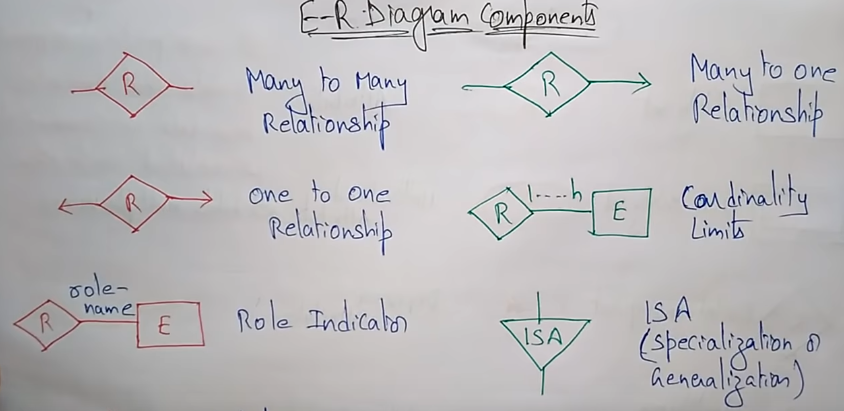


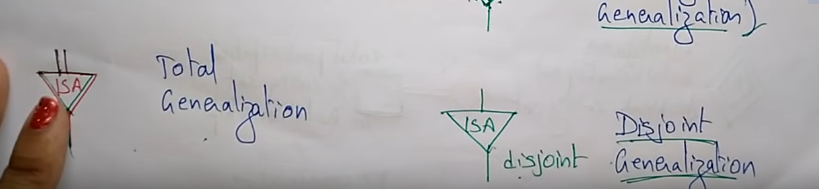


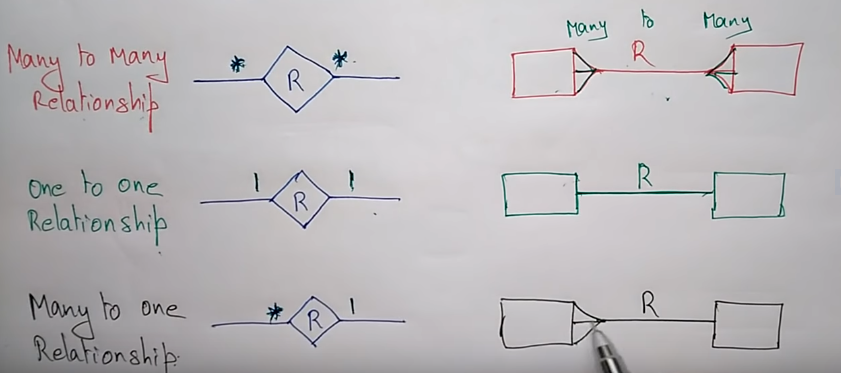


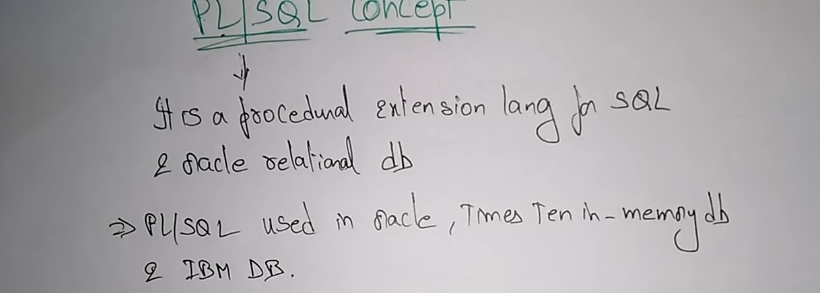


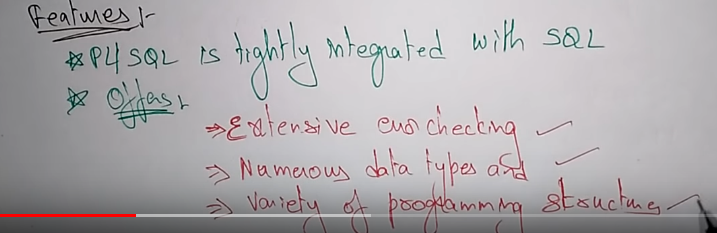


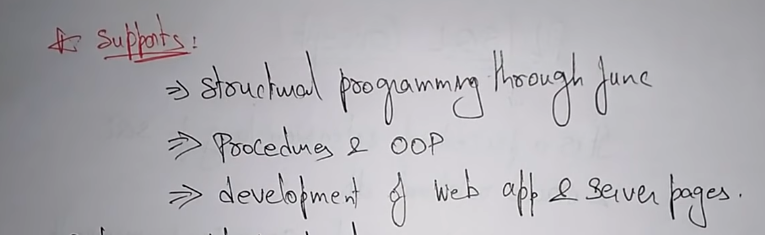


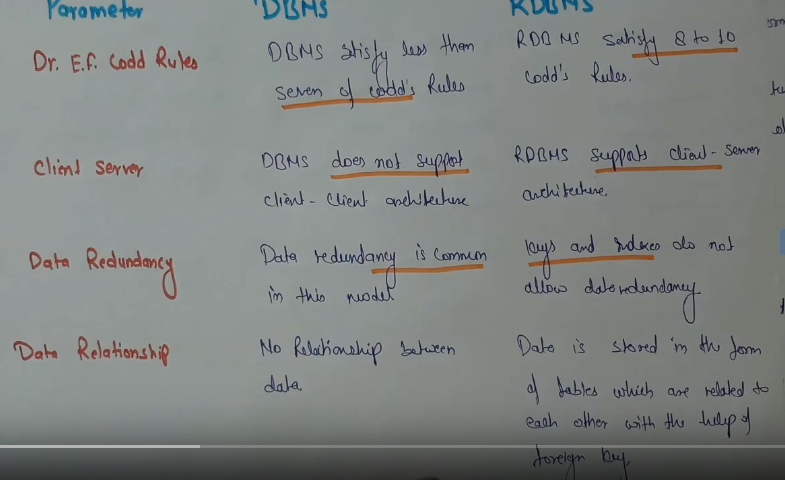


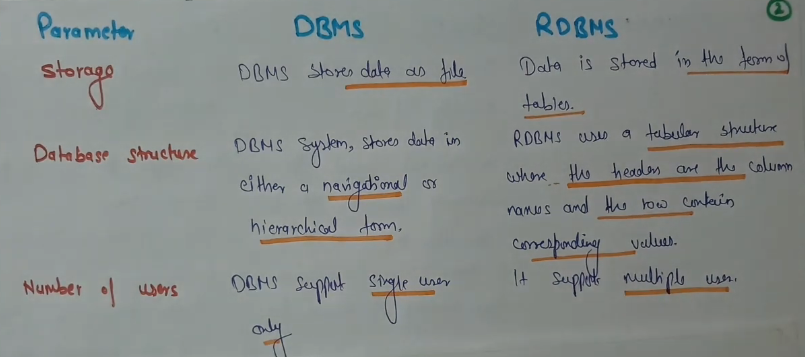


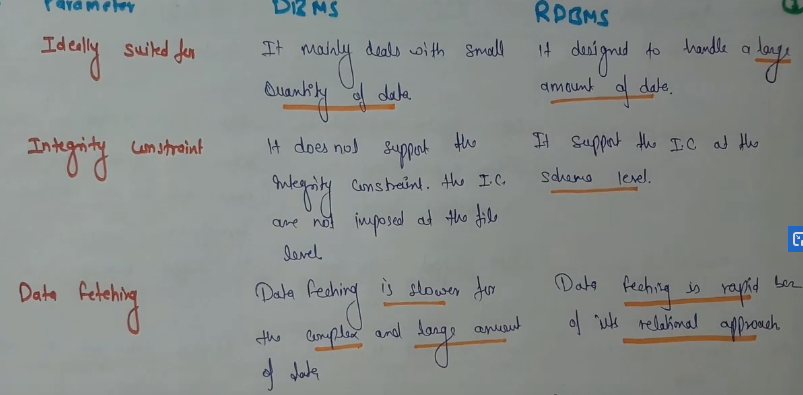


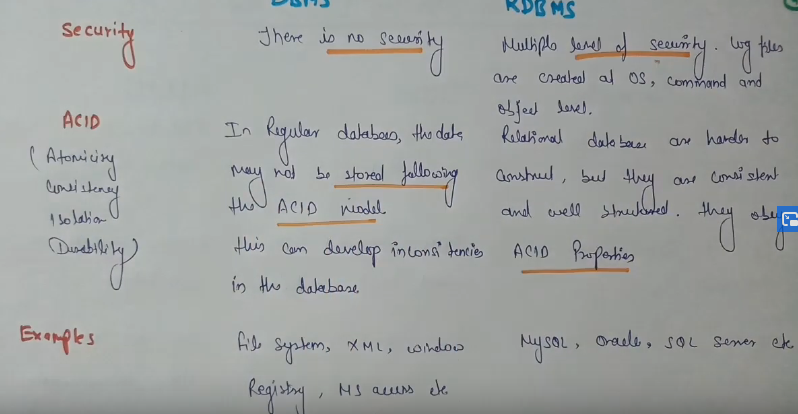












|  |  |  |
| --- | --- | --- |
| S.NO | High Level Language | Low Level Language |
| 1. | It is programmer friendly language. | It is a machine friendly language. |
| 2. | High level language is less memory efficient. | Low level language is high memory efficient. |
| 3. | It is easy to understand. | It is tough to understand. |
| 4. | It is simple to debug. | It is complex to debug comparatively. |
| 5. | It is simple to maintain. | It is complex to maintain comparatively. |
| 6. | It is portable. | It is non-portable. |
| 7. | It can run on any platform. | It is machine-dependent. |
| 8. | It needs compiler or interpreter for translation. | It needs assembler for translation. |
| 9. | It is used widely for programming. | It is not commonly used now-a-days in programming. |

**Examples of  high level language**

Following are all examples of a high level language. To convert them into machine-language, compilers are used.

* COBOL
* [JAVA](https://www.3ritechnologies.com/course/core-java-classes-in-pune/)
* C
* C++

**Examples of  low level language**

There are only two types of low level language are-

* Assembly Language
* Machine Code.