Ranganathan Institute of Library and Information Science, Patna
(A Complete solution of Library science & Software)
Registered from Govt. of India, Reg. No. Udyam-BR-26-0080476

Relational Database

By Afroz Ahmad, KBOPL, Mob. 9006658740

रिलेशनल डेटाबेस की शुरुआत 1970 में IBM Research Laboratory में एडगर फ्रैंक कॉड (Edgar Frank Codd) के द्वारा की गई। यह (1923-2003) आईबीएम के लिए काम करने वाले एक अंग्रेजी कंप्यूटर वैज्ञानिक थे, उन्होंने रिलेशनल डेटाबेस मैनेजमेंट सिस्टम का आविष्कार किया और 'रिलेशनल डेटाबेस के लिए 12 नियम' दिए। यह डेटाबेस बहुत ही आसान, शक्तिशाली, लचीला, और प्राकृतिक है। रिलेशनल डेटाबेस तालिकाओं (Tables) का एक सेट है जिसमें से डेटा को अलग अलग तरीकों से पुनःप्राप्त किया जा सकता है। यह संरचना टेबल (Structure table) जैसा होता है और टेबल को डेटाबेस की भाषा में रिलेशन कहते हैं इसलिए इसका नाम भी रिलेशनल डेटाबेस मॉडल है। इसमें पंक्तियाँ और स्तंभ (Rows and column) होते हैं। इस प्रकार के डेटाबेस में पंक्तियाँ (Rows) को रिकॉर्ड और स्तंभ (Column) को फील्ड कहते हैं। इसमें एक की (Key) होता है जिसमें टेबल को आपस में जोड़ा जाता है। जिस प्रकार हम लोग एक्सेल या गूगल शीट बनाते हैं यह भी रिलेशनल डेटाबेस मॉडल है। उदहारण के लिए स्टूडेंट्स के रिकॉर्ड का एक टेबल है तो प्राइमरी की स्टूडेंट्स का रोल नंबर ही होगा। उदहारण इस प्रकार है।

Roll No.	Name	Address	Caste
1	Anjay Kumar	Patna	General
2	Ratan Kumar	New Delhi	OBC
3	Afroz Ahmad	Mumbai	General
4	Vijay Singh	Kolkata	SC

इस डेटाबेस में मुख्य रूप से तीन तत्व होते हैं।

- **Field** – किसी भी डेटाबेस में कॉलम को फील्ड कहते हैं।
- **Record** – डेटाबेस में मौजूद रो (Rows) को रिकॉर्ड कहते हैं।
- **Table** – फील्ड और टेबल को मिला कर एक पूरा टेबल बनता है।
  - ❖ एक रिलेशनल डेटाबेस में तालिका (Table) हर पंक्ति (Row) का एक यूनिक ID के साथ साथ एक रिकॉर्ड भी होता है किसे कि की (Key) कहा जाता है।
  - ❖ तालिका (Table) के कॉलम में डेटा के गुण (Attribute) होते हैं और प्रत्येक और प्रत्येक रिकॉर्ड में एक मान (Value) होते हैं जिससे डेटा के बीच सम्बन्ध स्थापित करना सां हो जाता है।
  - ❖ रिलेशनल डेटाबेस रख रखाव (Maintain) के लिए रिलेशनल डेटाबेस मैनेजमेंट सिस्टम (RDBMS) का उपयोग किया जाता है।

कुछ रिलेशनल डेटाबेस इस प्रकार हैं।

- MySQL
- ORACLE
- Microsoft SQL server
- PostGre SQL
- IBM Db2

कुछ क्लाउड आधारित रिलेशनल डेटाबेस इस प्रकार हैं।

- Google cloud SQL
- Oracle cloud

- Microsoft SQL server
- IBM Db2 on cloud
- AWS relational database service
- SQL Azure

रिलेशनल डेटाबेस कैसे काम करता है।

रिलेशनल डेटाबेस के द्वारा कई सम्बंधित तालिकाओं से डेटा या किसी भी सूचना को जोड़ कर काम किया जाता है। इसमें एक कुंजी (Key) होता है। इसे प्राथमिक कुंजी (Primary key) के रूप जाना जाता है और इसे ID, सीरियल नंबर, उपयोगकर्ता का नाम आदि हो सकता है।

इसमें हर डेटाबेस का एक दूसरे से रिलेशन हैं। उपयोगकर्ता रिलेशनल डेटाबेस का उपयोग अपनी ज़रूरत के अनुसार करता है। आजकल क्लाउड आधारित रिलेशनल डेटाबेस का उपयोग व्यापक रूप से हो रहा है ताकि उपयोगकर्ता कहीं से भी कभी भी उपयोग कर सके। इस प्रकार का डेटाबेस SQL server, Oracle, MySQL, IBMDB2 आदि है।

रिलेशनल डेटाबेस के लाभ (Advantage of relational database)

- **सटीकता या शुद्धता (Accuracy)**: इस डेटाबेस में सटीकता या शुद्धता (Accuracy) होती है। गलत रिपोर्ट पाने की संभावना नहीं होती है। यानि ज़रूरत के अनुसार यथातथ्य (Exact) डेटा प्राप्त किया जा सकता है।
- **लचीलापन (Flexibility)**: उपयोगकर्ता के लिए जटिल से जटिल सूचनाओं को खोजना आसान है यानि जिस प्रकार से भी हम रिपोर्ट देखना चाहते हैं या प्राप्त करना चाहते है उसे प्राप्त कर सकते हैं।
- **सहयोग (Collaboration)**: एक समय में एक से अधिक उपयोगकर्ता एक प्रकार के डेटा को पुनः प्राप्त या पहुँच बना सकते हैं।
- **विश्वसनीय (Trust)**: यह डेटा परिपक्व (Mature) होता है। इसे अच्छी तरह से समझा जा सकता है यानि इसके रिपोर्ट को बहुत आसानी से समझ सकते है।
- **सुरक्षा (Security)**: इसमें डेटाबेस के अन्दर तालिकाओं (Tables) के अन्दर डेटा को विशेष उपयोगकर्ताओं के द्वारा ही उपयोग की अनुमति देने के लिए सीमित किया जा सकता है यानि यह डेटा को सुरक्षा प्रदान करता है।
- **भण्डारण क्षमता (Storage capicity):** इसमें कुशल भण्डारण क्षमता होती है और इसके रिकॉर्ड को परिवर्तित करने कि आवशयकता नहीं होती है।
- **डुप्लीकेशन (Duplication):** इसमें डेटा के डुप्लीकेशन से बचना आसान हैं।
- **सम्पादन (Editing):** इसमें किसी भी प्रकार की सूचना को जोड़ना, संशोधित करना या हटाना आसान होता है।

रिलेशनल डेटाबेस हानि (Disadvantage of relational database)

- रिलेशनल डेटाबेस बनाने में लागत बहुत अधिक होती है।
- इस डेटाबेस का प्रदर्शन प्रेजेंटेशन तालिकाओं (Tables) पर निर्भर करता है। अगर तालिकाओं की संख्या आशिक है तो प्रश्नों का उत्तर धीमा होगा।
- इस प्रकार के डेटाबेस में बड़ी मात्र में मेमोरी की आवशयकता होती है।

रिलेशनल डेटाबेस कि विशेषताएँ (Features of RDBMS)

- इसमें मल्टी यूजर और मल्टी लेवल सिक्यूरिटी की विशेषताएँ होती हैं।
- इसमें तुरंत नए डेटाबेस बनाये और उसे आसानी से संशोधित किये जा सकते हैं।
- यह कई ऑपरेटिंग सिस्टम पर काम कर सकता है।
- यह मैनेजमेंट सिस्टम के एडमिनिस्ट्रेटर को पूरा नियंत्रण प्रदान करता है कि उन्हें कौन कौन एक्सेस कर सकता है।
- यह इंडेक्स्ड फाइल की संरचना (Structure) प्रदान करता है।
- इसमें रो (Row) और कॉलम (Column) से युक्त तालिकाओं (Tables) के संग्रह के अन्दर डेटा का संगठन होता है।

रिलेशनल डेटाबेस मैनेजमेंट सिस्टम में कॉड के बारह नियम तेरह नियमों का एक समूह हैं (शून्य से बारह की संख्या)।

नियम 0: बुनियादी नियम (The foundation rule): किसी भी प्रणाली के लिए जिसे रिलेशनल डेटाबेस प्रबंधन प्रणाली के रूप में विज्ञापित किया जाता है, या दावा किया जाता है, उसे अपनी रिलेशनल क्षमताओं के माध्यम से डेटाबेस को पूरी तरह से प्रबंधित करने में सक्षम होना चाहिए।

कोई भी सिस्टम अपने आपको तभी RDBMS कह सकता है जब तक कि वह सभी चीज़ों को सम्बन्ध (relation) के रूप में न रखे। जैसे एक परिवार में सभी लोग एक दूसरे से जुड़े हुए होते हैं इसी प्रकार कोई भी डेटा बेस मैनेजमेंट सिस्टम (DBMS) बन रहा है तो एक प्रकार के डेटा को एक दुसरे से जुड़ा हुआ होना चाहिए उदहारण के लिए IRCTC का डेटाबेस ट्रेन से सम्बंधित है। इसमें ट्रेन के पैसेंजर की सूचना इसके मार्ग की सूचना, इसके कर्मचारियों कि सूचना एक दुसरे से जुड़ा हुआ होता है। इस सिस्टम में जो कुछ भी होगा एक दुसरे से जुदा हुआ होगा।

नियम 1: सूचना नियम (The information rule): एक संबंधपरक डेटाबेस में सभी सूचना स्पष्ट रूप से तार्किक स्तर पर और तालिका (Table) में मूल्यों द्वारा बिल्कुल एक तरह से दर्शायी जाती है और टेबल को ही रिलेशन कहते हैं।

किसी भी सिस्टम के आर्किटेक्चर की में तीन लेवल होते हैं।

i. **एक्सटर्नल लेवल**: इस लेवल में यूजर होता है।
ii. **लॉजिकल लेवल**: इस लेवल में प्रोग्राम की संरचना (Structure) होती है।
iii. **फिजिकल लेवल**: इस लेवल पर सभी डेटा होता है।

नियम 2: गारंटीकृत पहुंच नियम (The guaranteed access rule): रिलेशनल डेटाबेस में हमने जो भी डेटा टेबल के रूप में रखा या संग्रह किया है वह सभी डेटा के फील्ड पुनः प्राप्त (Accessible) करने योग्य होने चाहिए और उस तक पहुँच बनाने के लिए टेबल का नाम, प्राइमरी की और कॉलम का नाम ज़रूरी है और यह सभी प्राइमरी की पर निर्भर करता है।

नियम 3: शून्य मानों का व्यवस्थित उपचार (Systematic treatment of null values): नल वैल्यू का अर्थ शुन्य होता है यहाँ पर किसी भी खली कॉलम यानि उस कॉलम में डेटा नहीं होता है उसे नल वैल्यू कहते हैं। डेटाबेस में शुन्य वैल्यू को बहुत ध्यान से व्यव्हार (Treat) में लाना है। जब भी हम डेटाबेस बनाते है तो कोशिश करनी चाहिए कि डेटाबेस में शुन्य वैल्यू को न छोड़ें। उसमें कुछ न कुछ भरने की कोशिश करें। अगर इसमें भरने के लिए हमारे पास डेटा उपलब्ध नहीं है तो डमी डेटा को ही भर देना चाहिए।

नियम 4: संबंधपरक मॉडल पर आधारित डायनामिक ऑनलाइन कैटलॉग (Dynamic online catalog based on the relational model): यह नियम डेटा डिक्शनरी के बारे में है। इसमें डेटा के बारे में डेटा यानि मेटाडेटा को भी संग्रह करना ज़रूरी है, टेबल का नाम, प्राइमरी की किसे बनाया है यानि टेबल से सम्बंधित डेटा को भी संग्रह करना ज़रूरी है इसे ही हम कैटलॉग या डेटा डिक्शनरी कहते हैं। डेटा डिक्शनरी को भी वैसे ही संग्रह करना है जैसे सामान्य सूचना (Normal information) को टेबल के रूप में संग्रह करते हैं उसी प्रकार डेटा डिक्शनरी को भी टेबल के रूप में संग्रह करना होता है।

नियम 5: व्यापक डेटा उपभाषा नियम (The comprehensive data sublanguage rule): रिलेशनल डेटाबेस कई भाषा में सहायक (Support) सिद्ध होता है मगर एक भाषा ऐसी होनी चाहिए जिसमें डेटा का सभी काम हम कर सकें। उदहारण के लिए SQL (Structured Query language) का उपयोग सिस्टम को बनाने (Create), एडिट करने के लिए किया जाता है और हम उसमें उल्लेख कर सकें कि इसे कौन कौन एक्सेस कर सकता और कौन कौन एक्सेस नहीं कर सकता है क्या एक्सेस कर सकता है और क्या एक्सेस नहीं कर सकता है क्या यह ऑनलाइन या ऑफलाइन लेन देन (Transactions) हो सकता है। इसके अतिरिक्त इसमें निम्न काम हो सकते हैं।

i. Data definition: किसी एक भाषा के द्वारा डेटा को परिभाषित (Define) करें।
ii. View definition: किसी एक भाषा के द्वारा डेटा को देख सकें।

iii. Data manipulation by program: किसी एक भाषा के द्वारा डेटा को जोड़, डिलीट कर सकें।

iv. Integrity constraints: किसी एक भाषा के द्वारा डेटा का एक दुसरे से जुडाव होना चाहिए।

v. Authorization: कौन डेटा तक पहुँच बना सकता है और कौन डेटा तक पहुँच नहीं बना सकता है इसका प्राधिकरण (Authorization) होना चाहिए।

vi. Transaction boundaries (begin, commit and rollback): किसी एक भाषा के द्वारा अदन प्रदान, लेन देन सुविधाएँ होनी चाहिए।

नियम 6: दृश्य अद्यतन नियम (The view updating rule): जिस इस नियम में हम टेबल को महत्त्व देते है उसी प्रकार देखने को (Views) को भी महत्व देना चाहिए। जिस प्रकार टेबल को बनाने की प्रक्रिया, एडिट करने कि प्रक्रिया करते है ठीक उसी प्रकार देखने को भी बनाने, एडिट करने और बदलने कि प्रक्रिया भी कर सकें।

नियम 7: जोड़ने, अद्यतन करना और एडिट करना संभव हो (Possible for high-level insert, update and delete): जो भी डेटाबेस हम बना रहें हैं उसमें जोड़ने, अपडेट, डिलीट और एडिट करने की सुविधाएँ होनी चाहिए। इसे ही डेटा मैनीपुलेशन (Data manipulation) कहते हैं।

नियम 8: भौतिक डेटा स्वतंत्रता (Physical data independence): भौतिक डेटा में यह स्वतंत्रता होनी चाहिए कि अगर हम डेटाबेस में डेटा या डेटा का स्थान कुछ भी बदल रहें हैं तो इसका असर एक्सटर्नल लेवल (External level) यानि यूजर की खोज या पुनः प्राप्ति पर नहीं होना चाहिए।

नियम 9: तार्किक डेटा स्वतंत्रता (Logical data independence): तार्किक डेटा में भी यह प्रावधान (Provision) होनी चाहिए कि अगर स्कीमा या लॉजिकल लेवल पर कॉलम या टेबल में कुछ बदल रहें हैं तो तो इसका असर एक्सटर्नल लेवल (External level) यानि यूजर की खोज या पुनः प्राप्ति पर नहीं होना चाहिए।

नियम 10: अखंडता स्वतंत्रता (Integrity independence): अखंडता बाधा (Integrity constraint) नियमों का एक समूह है। इसका उपयोग सूचना की गुणवत्ता बनाए रखने के लिए किया जाता है। अखंडता बाधा (Integrity constraint) यह सुनिश्चित करती है कि डेटा प्रविष्टि, अद्यतनीकरण और अन्य प्रक्रिया इस तरह से की जानी चाहिए कि डेटा अखंडता (Data integrity) प्रभावित न हो यानि जब डेटा को किसी भो सॉफ्टवेर में संग्रह किया जाता है तो उसी संग्रह के नियम को Integrity constraint कहते हैं। अखंडता बाधा का उपयोग डेटाबेस को आकस्मिक क्षति से बचाने के लिए किया जाता है।

अखंडता स्वतंत्रता **(Integrity independence)** में भी स्वतंत्रता होनी चाहिए। एप्लीकेशन प्रोग्राम में हम किसी प्रकार का निर्देश नहीं देंगे बल्कि सभी इनटिगरिटी कंसट्रेंट SQL भाषा लगाएगा। जब कोई भी डेटा बकेंड (Backend) में रखते हैं वह SQL के अन्दर ही रखते हैं। हम NULL या foreign key सभी SQL में बनाये जाते हैं न कि एप्लीकेशन प्रोग्राम में।

नियम 11: वितरण स्वतंत्रता (Distribution independence): अगर हमारे पास बहुत बड़ा डेटा है तो इसे कई जगह पर रखा जाता है मगर यूजर से इसे छुपा दिया जाता है यानि यूजर को यह डेटा नहीं दिखना चाहिए।

नियम 12: The non subversion rule: हम जिस प्रोग्रामिंग भाषा में काम करते हैं वह हाई लेवल लैंग्वेज होती है मगर जब हम इसे कम्पाइल (Compile) करते हैं तो यह लो लैंग्वेज में बदल जाती है। ऐसा नहीं होना चाहिए कि जब हम हाई लेवल लैंग्वेज से लो लैंग्वेज में बदल रहें हों तो किसी प्रकार का कमियां आ जाये ऐसे बिलकुल भी नहीं होना चाहिए कि इनटिगरिटी कंसट्रेंट का उल्लंघन कर रहें हों।

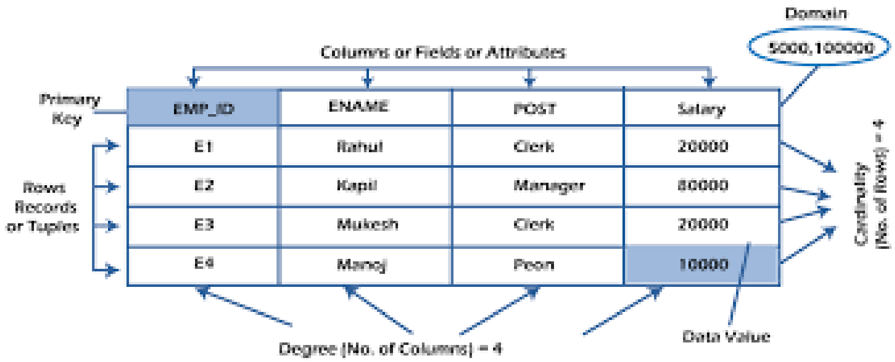

Image: Relational Database

इस प्रकार हम कह सकते हैं कि रिलेशनल डेटाबेस की ज़रूरत हर सॉफ्टवेर में होती है। पुस्तकालय के लिए बनाई गई सॉफ्टवेर इ-ग्रन्थालय और अन्य सॉफ्टवेर में भी रिलेशनल डेटाबेस का उपयोग किया जाता है ताकि हम पुस्तकालय से सम्बंधित किसी प्रकार की रिपोर्ट को आसानी से प्राप्त कर सकें।

Afroz Ahmad
Khuda Bakhsh O. P. Library, Patna
Mob. 9006658740

RANGANATHAN INSTITUTE OF LIBRARY & INFORMATION SCIENCE, PATNA
RI LIS